

Universidad de Cuenca

Facultad de Ingeniería

Escuela de Informática



“EVALUACIÓN DEL RENDIMIENTO ACADÉMICO DE LOS  
ESTUDIANTES DE LA ASIGNATURA DE ALGORITMOS, DATOS Y  
ESTRUCTURAS DE LA FACULTAD DE INGENIERÍA A PARTIR DE LA  
CONSTRUCCIÓN DE UN MODELO CONCEPTUAL DE DATOS  
APLICADO A UN MOOC.”

Autores:

Andrea Carolina Peralta Bravo

Christian Patricio Piedra Orellana

Director:

Ing. Jorge Javier Maldonado Mahauad, Msc.

Tesis de grado previa a la obtención del Título de  
Ingeniero de Sistemas

2014



# RESUMEN

Desde el año 2008, la educación en línea ha experimentado un auge significativo gracias a la creación de un nuevo paradigma conocido actualmente como MOOC (Massive Open Online Course). Los MOOCs han tenido gran aceptación a nivel mundial gracias a sus principales características: abierto, colaborativo, masivo y gratuito.

La presente tesis sintetiza el resultado de la evaluación sobre el rendimiento académico de los estudiantes de la asignatura de Algoritmos, Datos y Estructuras de la Facultad de Ingeniería de la Universidad de Cuenca. Para el análisis de los datos relacionados con el rendimiento académico, se diseñó e implementó en un MOOC los contenidos de los 3 primeros capítulos de la asignatura y que fueron tomados por los estudiantes de la asignatura. La plataforma con la que se desplegó el MOOC registró la interacción de los estudiantes con el material y a partir de estos, se construyó un modelo de datos para evaluar e interpretar los datos.

En el desarrollo de este proyecto de tesis, se implementó un modelo de MOOC que abarca los tres primeros capítulos de la asignatura antes mencionada, se evalúa en un universo de 140 estudiantes, de los cuales 59 alumnos utilizaron el MOOC como complemento a las clases presenciales, donde en la casa se revisaba la parte conceptual y en clases se hacía exclusivamente práctica; mientras que 81 alumnos tomaron la asignatura siguiendo la clase magistral.

Para contrastar la efectividad del MOOC dentro de una estrategia de aprendizaje presencial, se contrastó estos resultados con los de los estudiantes que tomaron la asignatura sin este tipo de mediadores del aprendizaje.

Como resultado de la evaluación se puede concluir que el modelo conceptual permite obtener una representación de datos independientes del entorno físico, lo que facilita la exportación del mismo a diferentes casos de estudio y que el uso del MOOC como mediadores de los contenidos permite que los estudiantes puedan reforzar de forma autónoma los temas revisados en clases generando un mejor rendimiento académico.

**Palabras Clave:** MOOC, Educación en Línea, Algoritmos, Datos, Estructuras, Google Course Builder, Python, Data Warehouse, Educational Data Mining.



# ABSTRACT

Since 2008, online education has boomed thanks to significant training paradigm called MOOC. MOOCs have been widely accepted around the world thanks to their main features: open, collaborative, massive and free. In our country, unfortunately, the use of MOOC as academic resource is limited.

This thesis includes the result about the evaluation of students' academic performance into the subject Algorithms, Data and Structures offered by Engineering School from University of Cuenca which emphasizes the design, implementation and dissemination at a MOOC. Also, there is an analysis of results by developing a data conceptual model which use generated data into the MOOC.

In the development of this thesis project, a MOOC model is implemented which include the first three chapters of the mentioned subject, where it is evaluated based on a universe of 59 students who take the MOOC as a complement with normal classes, this versus 81 students that take the subject with master class.

The objective of this work was to verify if MOOCs could be used as support for information transmission to students as a master class. Such that, knowledge assimilation was made by doing practical exercises, changing teacher's role from knowledge transmitter to academic tutor by this proposal. At the end the analyses indicated that the MOOC is a valuable tool in education.

**Keywords:** MOOC, Online Education, Algorithms, Data, Structures, Google Course Builder, Python, Data Warehouse, Educational Data Mining.

# Contenido

RESUMEN .....	2
ABSTRACT .....	3
Agradecimientos .....	17
Dedicatoria .....	18
Dedicatoria .....	19
CAPÍTULO 1: Introducción .....	20
1.1 Panorama general .....	20
1.2 Planteamiento del problema y justificación.....	20
1.3 Objetivos del trabajo .....	22
1.3.1 Objetivo General .....	22
1.3.2 Objetivos específicos .....	22
1.4 Metodología de la investigación .....	23
1.5 Estructura de la tesis .....	23
CAPÍTULO 2: Massive Open Online Courses (MOOC) .....	25
2.1 Introducción .....	25
2.2 ¿Qué es un MOOC? .....	25
2.3 ¿Para qué sirve un MOOC? .....	26
2.4 Características de un MOOC .....	27
2.5 Taxonomía de MOOC .....	31
2.6 Plataformas MOOC.....	34
2.7 Ventajas y desventajas de los MOOC.....	37
2.8 Recapitulación .....	39
CAPÍTULO 3: Plataformas para MOOC .....	41
3.1 Introducción .....	41
3.2 Plataformas para MOOC .....	42
3.2.1 Google Course Builder.....	48
3.2.2 COURSERA .....	50
3.2.3 Miríada X .....	53
3.2.4 EdX.....	56
3.2.5 OpenCourseWare (OCW) .....	58
3.3 Diagramas de clase de plataformas MOOC.....	60
3.4 Cuadro comparativo entre plataformas MOOC .....	65
3.5 Recapitulación .....	71
CAPÍTULO 4: Diseño e Implementación de un MOOC .....	72





4.1 Introducción .....	72
4.2 Metodología para la producción de un MOOC .....	72
4.2.1 Materia a impartirse y personas a las que está orientado.....	72
4.2.2 Objetivos del curso.....	73
4.2.3 Equipo docente .....	73
4.2.4 Estructura del curso .....	73
4.2.5 Participación de los estudiantes.....	83
4.2.6 Evaluación .....	83
4.2.7 Herramientas de comunicación .....	85
4.2.8 Calendario .....	85
4.3 Implementación del MOOC.....	85
4.3.1 Documentación técnica.....	85
4.3.1.1 HTML .....	85
4.3.1.2 CSS.....	86
4.3.1.3 JAVASCRIPT .....	86
4.3.1.4 PYTHON .....	86
4.3.1.5 GOOGLE APP ENGINE .....	86
4.3.1.6 GOOGLE BIGTABLE .....	87
4.3.2 Instalación y configuración de las herramientas de desarrollo de Course Builder....	87
4.3.2.1 Instalación de Google App Engine.....	87
4.3.2.2 Configuración del archivo course.yaml .....	89
4.3.2.3 Configuración del archivo course_template.yaml .....	90
4.3.2.4 Creación de aplicación en App Engine. ....	91
4.3.2.5 Despliegue del MOOC en App Engine .....	92
4.3.2.6 Configuración del MOOC desde la nube .....	93
4.3.2.7 Creación de Unidades .....	94
4.3.2.8 Creación de Lecciones .....	94
4.3.2.9 Creación de actividades .....	96
4.3.2.10 Creación de evaluaciones.....	96
4.3.2.11 Modificar la presentación del MOOC.....	98
4.3.2.12 Cambiar las opciones de configuración del curso .....	100
4.3.2.13 Creación del cuestionario para registro de estudiantes .....	102
4.3.2.14 Creación de anuncios para el MOOC .....	102
4.3.2.15 Obtención de datos de los estudiantes registrados en el MOOC .....	103
4.4 Recapitulación .....	104

CAPÍTULO 5: Diseño e Implementación de un Data Warehouse para análisis de resultados del MOOC.....	105
5.1 Introducción .....	105
5.2 Fundamentación Teórica.....	105
5.2.1 Data Warehouse.....	105
5.2.2 Modelos multidimensionales .....	107
5.2.3 Descripción de metodologías para Data Warehouse.....	107
5.2.3.1 Metodología Kimball .....	107
5.2.3.2 Metodología CRISP .....	108
5.2.3.3 Metodología HEFESTO .....	109
5.3 Selección de la metodología .....	111
5.4 Aplicación de la metodología HEFESTO .....	112
5.4.1 Análisis de requerimientos.....	112
5.4.1.1 Identificar preguntas .....	112
5.4.1.2 Identificar indicadores y perspectivas.....	113
5.4.1.3 Modelo Conceptual .....	115
5.4.2 Análisis de fuentes de datos.....	115
5.4.2.1 Conformar Indicadores.....	115
5.4.2.2 Establecer correspondencias .....	116
5.4.2.3 Modelo Conceptual Ampliado .....	121
5.4.3 Modelo lógico del Data Warehouse.....	122
5.4.3.1 Tipo del modelo lógico del Data Warehouse .....	123
5.4.3.2 Uniones .....	126
5.4.4 Integración de datos .....	128
5.4.4.1 Carga Inicial .....	128
5.5 Recapitulación .....	128
CAPÍTULO 6: Caso de estudio .....	131
6.1 Introducción .....	131
6.2 Selección y definición del caso de estudio .....	131
6.2.1 Ámbitos en los que es relevante el estudio .....	131
6.2.2 Justificación .....	132
6.2.3 Objetivo del caso de estudio .....	132
6.3 Fundamentación teórica .....	132
6.3.1 Estilos de Aprendizaje .....	132
6.3.2 Minería de datos en la educación .....	133
6.4 Elaboración de preguntas .....	134



6.5 Localización de las fuentes de datos .....	135
6.6 Diseño y Metodología .....	135
6.7 Análisis de datos.....	137
6.7.1 Respuesta a preguntas del caso de estudio .....	137
6.7.2 Aplicación de minería de datos educacional.....	144
6.7.2.1 Clasificación de un tipo de estudiante .....	144
6.7.2.2 Clúster de un tipo de estudiante.....	149
6.8 Interpretación de Resultados.....	153
6.9 Recapitulación .....	155
CAPÍTULO 7: Conclusiones y trabajos futuros.....	156
7.1 Conclusiones.....	156
7.2 Líneas de trabajo futuras.....	157
7.3 Lecciones Aprendidas.....	159
Referencias.....	160
ANEXOS .....	167
ANEXO 1: CONTENIDO Y ACTIVIDADES DEL MOOC .....	167
ANEXO 2: BANCO DE PREGUNTAS PARA EVALUACIONES.....	203
ANEXO 3: Herramientas para la implementación de un data warehouse.....	209
Procesos ETL.....	209
Pentaho Data Integration (Kettle).....	210
JSON .....	210
EXCEL.....	210
BASES DE DATOS .....	211
ANEXO 4: HERRAMIENTAS TECNOLÓGICAS UTILIZADAS PARA EL CASO DE ESTUDIO .....	212
Pentaho BI-Server .....	212
Configuración de Pentaho BI-SERVER .....	212
Weka .....	215

## Índice de figuras

Figura 2.1: Infraestructura tecnológica para un MOOC. ....	28
Figura 2.2: Elementos de un MOOC. ....	30
Figura 2.3: Tipos de MOOC según la teoría de aprendizaje. ....	33
Figura 2.4: Tipos de MOOC según el tipo de plataforma. ....	34
Figura 2.5: Línea de Tiempo de los MOOCs. ....	37
Figura 3.1: Mapa Geográfico de los países en donde se utiliza Google Course Builder. ....	42
Figura 3.2: Mapa Geográfico de los países en donde se utiliza EdX. ....	42
Figura 3.3: Mapa Geográfico de los países en donde se utiliza Coursera. ....	43
Figura 3.4: Mapa Geográfico de los países en donde se utiliza Miriada X. ....	43
Figura 3.5: Mapa Geográfico de los países en donde se utiliza Open Course Ware. ....	44
Figura 3.6: Mapa Geográfico de los países en donde se utiliza Udacity. ....	44
Figura 3.7: Mapa Geográfico de los países en donde se utiliza Eliademy. ....	45
Figura 3.8: Mapa Geográfico de los países en donde se utiliza Open Class. ....	45
Figura 3.9: Mapa Geográfico de los países en donde se utiliza OpenMOOC. ....	46
Figura 3.10: Porcentaje de incidencias de plataformas MOOC en el mundo. ....	47
Figura 3.11: Tiempo en el mercado de las plataformas MOOC. ....	48
Figura 3.12: Logotipo Course Builder. ....	48
Figura 3.13: Arquitectura de Google Course Builder. ....	49
Figura 3.14: Logotipo de Coursera. ....	50
Figura 3.15: Página principal de los cursos ofertados en Coursera. ....	50
Figura 3.16: Página de Coursera para obtener certificados de MOOC aprobados. ....	52
Figura 3.17: Logotipo de MiriadaX. ....	53
Figura 3.18: Universidades asociadas a Miriada X. ....	54
Figura 3.19: Certificado gratuito emitido por MiriadaX. ....	55
Figura 3.20; Certificado de pago emitido por MiriadaX. ....	55
Figura 3.21: Logotipo de edX. ....	56
Figura 3.22: Arquitectura de la plataforma EDx. ....	57
Figura 3.23: Logotipo de Open Course Ware. ....	58
Figura 3.24: Diagrama de clases de un curso en Google Course Builder. ....	60
Figura 3.25: Diagrama de clases de un curso en Coursera. ....	61
Figura 3.26: Diagrama de clases de un curso en MiriadaX. ....	62
Figura 3.27: Diagrama de clases de un curso en EdX. ....	63
Figura 3.28: Diagrama de clases de un curso en Open Course Ware. ....	64
Figura 4.1: Página de Inicio del MOOC. ....	77
Figura 4.2: Aviso de Bienvenida al MOOC. ....	78
Figura 4.3: Foro del MOOC. ....	78
Figura 4.4: Cuestionario de registro de estudiantes en el MOOC. ....	79
Figura 4.5: Progreso del estudiante en el MOOC. ....	79
Figura 4.6: Contenidos del MOOC. ....	80
Figura 4.7: Estructura de las Unidades del MOOC. ....	80
Figura 4.8: Estructura de las unidades. ....	81
Figura 4.9: Menú de lecciones y actividades de la unidad. ....	81
Figura 4.10: Estructura de las lecciones del MOOC. ....	82
Figura 4.11: Estructura de las actividades del MOOC. ....	83
Figura 4.12: Estructura de las evaluaciones. ....	84
Figura 4.13: Pantalla principal de Google App Engine. ....	88



Figura 4.14: Estructura de archivos de Google Course Builder.....	88
Figura 4.15: Agregar un proyecto de GCB en GAE. ....	89
Figura 4.16: Configuración del archivo course.yaml. ....	90
Figura 4.17: Configuración del archivo course_template.yaml. ....	90
Figura 4.18: Pantalla principal de App Engine.....	91
Figura 4.19: Pantalla para crear una aplicación en App Engine. ....	92
Figura 4.20: Configuración del archivo app.yaml.....	92
Figura 4.21: Configuración del despliegue de Google Course Builder. ....	92
Figura 4.22: Menú principal del MOOC para la cuenta de administrador. ....	93
Figura 4.23: Dashboard del MOOC.....	93
Figura 4.24: Configuración de unidades.....	94
Figura 4.25: Configuración de las lecciones. ....	95
Figura 4.26: Estructura de actividad en Course Builder.....	96
Figura 4.27: Configuración de evaluaciones en Google Course Builder. ....	97
Figura 4.28: Estructura de evaluación en Course Builder. ....	98
Figura 4.29: Archivo butterbar.css. ....	99
Figura 4.30: Archivo main.css.....	99
Figura 4.31: Configuración del curso desde el dashboard. ....	100
Figura 4.32: Configuración del curso desde el dashboard. ....	101
Figura 4.33: Código para la creación de registro de estudiantes.....	102
Figura 4.34: Configuración de avisos para el MOOC.....	103
Figura 4.35: Datos de Script para descarga de información. ....	103
Figura 4.36: Comando para descarga de datos del MOOC. ....	104
Figura 5.1: Procesos del Data Warehouse. ....	106
Figura 5.2: Ejemplo de modelo multidimensional. ....	107
Figura 5.3: Metodología de Kimball. ....	108
Figura 5.4: Metodología CRISP.....	109
Figura 5.5: Metodología HEFESTO. ....	110
Figura 5.6: Modelo Conceptual.....	115
Figura 5.7: Estructura del archivo JSON obtenido del MOOC. Fuente: Elaboración propia ....	117
Figura 5.8: Estructura de la encuesta aplicada a los estudiantes que no cursaron el MOOC...	118
Figura 5.9: Estructura de las notas de los estudiantes.....	118
Figura 5.10: Correspondencia de los indicadores y las fuentes de datos. ....	119
Figura 5.11: Correspondencia de perspectivas y las fuentes de datos. ....	120
Figura 5.12: Modelo conceptual expandido. ....	122
Figura 5.13: Tabla Dimensión Edad.....	123
Figura 5.14: Tabla Dimensión Ciudad.....	123
Figura 5.15: Tabla Dimensión Actividades. ....	123
Figura 5.16: Tabla Dimensión ResultadoMOOC.....	124
Figura 5.17: Tabla Dimensión EvaluacionTeorica.....	124
Figura 5.18: Tabla Dimensión EvaluacionPractica.....	124
Figura 5.19: Tabla Dimensión ResultadoEvaluacion. ....	124
Figura 5.20: Tabla Dimensión EstiloPredominante. ....	125
Figura 5.21: Tabla Dimensión EstiloSecundario. ....	125
Figura 5.22: Tabla Dimensión Estado. ....	125
Figura 5.23: Tabla Dimensión Sexo. ....	125
Figura 5.24: Tabla Dimensión Docente. ....	126

Figura 5.25: Tabla Dimensión ConocimientoPrevio.....	126
Figura 5.26: Tabla Dimensión TipoAlumno.....	126
Figura 5.27: Tabla de hechos Alumno.....	127
Figura 5.28: Modelo lógico de Alumno.....	127
Figura 5.29: Transformación en Pentaho Data-Integration.....	129
Figura 5.30: Estructura del cubo multidimensional del BI-Server.....	130
Figura 6.1: Formato de Encuesta a alumnos sin MOOC.....	136
Figura 6.2: Reporte Muestra-EstiloAprendizaje.....	137
Figura 6.3: Gráfico del reporte Muestra-EstiloAprendizaje.....	138
Figura 6.4: Reporte Muestra-EstiloAprendizaje-Sexo.....	139
Figura 6.5: Reporte Muestra-Resultados.....	140
Figura 6.6: Reporte EstiloAprendizaje-Evaluación.....	140
Figura 6.7: Gráfico reporte Estilo-Aprendizaje.....	140
Figura 6.8: Reporte Estado-ConocimientoPrevio.....	141
Figura 6.9: Reporte Evaluación teórico.....	141
Figura 6.10: Estadísticas del Reporte Evaluación Teórica.....	141
Figura 6.11: Reporte Evaluación Práctico.....	142
Figura 6.12: Estadísticas del Reporte Evaluación Práctico.....	142
Figura 6.13: Reporte EstiloAprendizaje-ResultadoMooc.....	142
Figura 6.14: Gráfico EstiloAprendizaje-ResultadoMooc.....	143
Figura 6.15: Reporte EstiloAprendizaje-Evaluación.....	143
Figura 6.16: Gráfico Reporte EstiloAprendizaje-Evaluación.....	143
Figura 6.17: Consulta SQL de los datos de entrada.....	144
Figura 6.18: Carga de datos en Weka.....	146
Figura 6.19: Configuración del algoritmo NaiveBayes.....	146
Figura 6.20: Resultado de datos analizados.....	147
Figura 6.21: Gráfico de datos analizados.....	147
Figura 6.22: Resumen después de aplicar el algoritmo NaiveBayes.....	147
Figura 6.23: Configuración del algoritmo J48.....	148
Figura 6.24: Gráfico de datos analizados.....	148
Figura 6.25: Resumen después de aplicar el algoritmo J48.....	149
Figura 6.26: Consulta SQL de los datos de entrada.....	149
Figura 6.27: Carga de datos en el Weka.....	150
Figura 6.28: Configuración del algoritmo FarthestFirst.....	151
Figura 6.29: Gráfico de clusterización con el algoritmo FasthestFirst.....	151
Figura 6.30: Resumen después de aplicar el algoritmo FarthestFirst.....	152
Figura 6.31: Configuración del algoritmo MakeDensityBasedClustered.....	152
Figura 6.32: Resultado después de aplicar el algoritmo MakeDensityBasedClusterer.....	153
Figura 6.33: Resumen después de aplicar el algoritmo MakeDensityBasedClusterer.....	154
Figura A1.1: Estructura de una computadora.....	167
Figura A1.2: Estructura de un programa.....	167
Figura A1.3: Distribución de un programa.....	168
Figura A1.4: Lenguajes de Programación.....	169
Figura A1.5: Pasos para realizar un programa.....	171
Figura A1.6: Visión general de un programa.....	173
Figura A1.7: Ejemplo de un algoritmo.....	173
Figura A1.8: Etapas de resolución de un problema.....	175

Figura A1.9: Ejemplo 1 de la representación de una variable. ....	178
Figura A1.10: Ejemplo 2 de la representación de una variable. ....	178
Figura A1.11: Ejemplo de sintaxis en Python. ....	179
Figura A1.12: Tipos de datos primitivos. ....	180
Figura A1.13: Simbología de los diagramas de flujo. ....	182
Figura A1.14: Ejemplo de algoritmo con diagramas de flujo. ....	183
Figura A1.15: Estándar de pseudocódigo. ....	184
Figura A1.16: Ejemplo de pseudocódigo. ....	185
Figura A1.17: Ejemplo de prueba de escritorio. ....	187
Figura A1.18: Secuencia de instrucciones. ....	188
Figura A1.19: Estructura de decisión. ....	189
Figura A1.20: Instrucciones en bloques. ....	190
Figura A1.21: Operadores de comparación. ....	191
Figura A1.22: Ejemplo de bloques if anidados y condiciones con operadores. ....	192
Figura A1.23: Ejemplo del uso de los operadores AND, OR, NOT. ....	192
Figura A1.24: Estructura if-else en pseudocódigo. ....	192
Figura A1.25: Estructura if-else anidados en pseudocódigo. ....	193
Figura A1.26: Estructura de repetición. ....	194
Figura A1.27: Ejemplo de algoritmo para tabla de multiplicar del 5. ....	194
Figura A1.28: Ejemplo de algoritmo utilizando estructuras de repetición. ....	195
Figura A1.29: Ejecución de algoritmo. ....	195
Figura A1.30: Uso de función range. ....	196
Figura A1.31: Uso de función range con valor inicial y final. ....	196
Figura A1.32: Uso de función range con incremento. ....	197
Figura A1.33: Uso de función range con decremento. ....	197
Figura A1.34: Pseudocódigo estructura FOR. ....	197
Figura A1.35: Pseudocódigo estructura FOR con función RANGE. ....	198
Figura A1.36: Estructura de Selección. ....	199
Figura A1.37: Estructura de selección. ....	200
Figura A1.38: Funcionamiento sentencia WHILE. ....	201
Figura A1.39: Ejemplo 1 de sentencia WHILE. ....	201
Figura A1.40: Ejemplo 2 de sentencia WHILE. ....	201
Figura A1.41: Pseudocódigo sentencia WHILE. ....	202
Figura A4.1: Pantalla inicial Pentaho BI-Server. ....	212
Figura A4.2: Pantalla de configuración de base de datos. ....	213
Figura A4.3: Selección de conexión. ....	213
Figura A4.4: Selección de tablas del Data Warehouse. ....	214
Figura A4. 5: Creación de uniones de tablas. ....	214
Figura A4.6: Pantalla de Finalización de creación de Data Source. ....	215



## Índice de tablas

Tabla 2.1: Comparación de Actividades de un MOOC y de una clase tradicional. ....	31
Tabla 3.1: Incidencias de empleo de plataformas MOOC en el mundo.....	46
Tabla 3.2: Tiempo en el mercado de las plataformas MOOC. Fuente: Elaboración propia.....	47
Tabla 3.3: Descripción de las características a evaluar en el enfoque “Elementos del curso” ...	65
Tabla 3.4: Descripción de las características a evaluar en el enfoque “Aspectos Técnicos” .....	66
Tabla 3.5: Cuadro comparativo de las características de las plataformas MOOC. ....	66
Tabla 3.6: Características y subcaracterísticas de la Norma ISO 25010:2010.....	69
Tabla 3.7: Descripción de las métricas utilizadas en la evaluación de las plataformas. ....	70
Tabla 3.8: Cuadro comparativo de las plataformas MOOC en base a las características y subcaracterísticas establecidas en la Norma ISO 25010:2010.....	70
Tabla 4.1: Roles del equipo docente. ....	73
Tabla 4.2: Estructura del MOOC.....	74
Tabla 4.3: Estructura de la unidad 1.....	75
Tabla 4.4: Estructura de la unidad 2.....	76
Tabla 4.5: Estructura de la unidad 3.....	76
Tabla 4.6: Puntaje de evaluación del MOOC.....	83
Tabla 4.7: Tiempo de dedicación al MOOC. ....	85
Tabla 5.1: Cuadro resumen de las metodologías.....	112
Tabla 6.1: Variables de entrada para clasificación.....	145
Tabla 6.2: Variables consideradas para clusterización.....	150
Tabla A1.1: Actividades de la lección 1 del módulo 1. ....	168
Tabla A1.2: Actividades de la lección 2 del módulo 1. ....	170
Tabla A1.3: Actividades de la lección 3 del módulo 1. ....	171
Tabla A1.4: Actividades de la lección 4 del módulo 1. ....	174
Tabla A1.5: Actividades de la lección 5 del módulo 1. ....	175
Tabla A1.6: Actividades de la lección 1 del módulo 2. ....	177
Tabla A1.7: Actividades de la lección 2 del módulo 2. ....	180
Tabla A1.8: Actividades de la lección 3 del módulo 2. ....	181
Tabla A1.9: Actividades de la lección 4 del módulo 2. ....	183
Tabla A1.10: Actividades de la lección 5 del módulo 2. ....	185
Tabla A1.11: Actividades de la lección 6 del módulo 2. ....	186
Tabla A1.12: Actividades de la lección 7 del módulo 2. ....	187
Tabla A1.13: Actividades de la lección 1 del módulo 3. ....	189
Tabla A1.14: Actividades de la lección 2 del módulo 3. ....	193
Tabla A1.15: Actividades de la lección 3 del módulo 3. ....	198
Tabla A1.16: Actividades de la lección 4 del módulo 3. ....	199
Tabla A1.17: Actividades de la lección 5 del módulo 3. ....	202
Tabla A2.1: Evaluación del módulo 1. ....	204
Tabla A2.2: Evaluación del módulo 2. ....	205
Tabla A2.3: Evaluación del módulo 3. ....	206
Tabla A2.4: Evaluación final .....	208





Universidad de Cuenca  
Clausula de derechos de autor

*Andrea Carolina Peralta Bravo*, autora de la tesis "Evaluación del rendimiento académico de los estudiantes de la asignatura de algoritmos, datos y estructuras de la Facultad de Ingeniería a partir de la construcción de un modelo conceptual de datos aplicado a un MOOC", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Ingeniera de Sistemas. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autora

Cuenca, 14 de noviembre de 2014

Andrea Carolina Peralta Bravo

C.I: 0104815956



Universidad de Cuenca  
Clausula de propiedad intelectual

---

*Andrea Carolina Peralta Bravo*, autora de la tesis "Evaluación del rendimiento académico de los estudiantes de la asignatura de algoritmos, datos y estructuras de la Facultad de Ingeniería a partir de la construcción de un modelo conceptual de datos aplicado a un MOOC", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autora.

Cuenca, 14 de noviembre de 2014

---

Andrea Carolina Peralta Bravo

C.I: 0104815956



Universidad de Cuenca  
Clausula de derechos de autor

---

*Christian Patricio Piedra Orellana*, autor de la tesis "Evaluación del rendimiento académico de los estudiantes de la asignatura de algoritmos, datos y estructuras de la Facultad de Ingeniería a partir de la construcción de un modelo conceptual de datos aplicado a un MOOC", reconozco y acepto el derecho de la Universidad de Cuenca, en base al Art. 5 literal c) de su Reglamento de Propiedad Intelectual, de publicar este trabajo por cualquier medio conocido o por conocer, al ser este requisito para la obtención de mi título de Ingeniero de Sistemas. El uso que la Universidad de Cuenca hiciere de este trabajo, no implicará afección alguna de mis derechos morales o patrimoniales como autor

Cuenca, 14 de noviembre de 2014



---

Christian Patricio Piedra Orellana

C.I: 0104633169



Universidad de Cuenca  
Clausula de propiedad intelectual

---

*Christian Patricio Piedra Orellana*, autor de la tesis "Evaluación del rendimiento académico de los estudiantes de la asignatura de algoritmos, datos y estructuras de la Facultad de Ingeniería a partir de la construcción de un modelo conceptual de datos aplicado a un MOOC", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 14 de noviembre de 2014

---

Christian Patricio Piedra Orellana

C.I: 0104633169



# Agradecimientos

El presente proyecto de tesis no hubiera sido posible sin la ayuda de nuestro director de tesis, Ing. Jorge Maldonado, que estuvo siempre dispuesto a guiarnos y apoyarnos de manera desinteresada e incondicional. Gracias Ingeniero por la confianza depositada en nosotros, por cada uno de los consejos y por haber sido un pilar fundamental para la realización de este proyecto.

Agradecemos también a los docentes de la asignatura de Algoritmos, datos y estructuras: Ing. Jorge Maldonado, Ing. Jorge Bermeo, Ing. Magali Mejía, Ing. Johnny Solorzano e Ing. Malhena Sánchez por brindarnos su ayuda para la realización del caso de estudio y confiar en el trabajo que realizamos.

Finalmente, agradecemos a todas las personas que de una u otra manera estuvieron con nosotros apoyándonos para cumplir con el objetivo de culminar este proyecto.

Andrea, Christian



# Dedicatoria

Este trabajo va dedicado en primer lugar a Dios por haberme bendecido y guiado mi camino durante toda mi vida.

A mis padres, Luis y Claudia, por ser un pilar fundamental en mi vida, por no dejarme caer cuando las cosas se ponían feas, por haber confiado en mí cuando estaba a punto de darme por vencida. Gracias por ser mi guía y fuerza para alcanzar este objetivo.

También quiero dedicar este trabajo a mis hermanos, Jorge, Gaby y Tefa, gracias por haberme apoyado durante todo este tiempo, por su todo su cariño y comprensión.

Finalmente, pero no por eso el menos importante, este trabajo va dedicado a mi esposo, Fernando, gracias por haber compartido todo este camino conmigo, gracias por cada palabra de aliento, gracias por ser mi compañero y mi amigo.

Andrea



# Dedicatoria

Primero que nada esta tesis la dedico a Dios quién ha venido bendiciéndome y dándome fortaleza durante todos estos años.

La dedico a toda mi familia en especial a mis padres Patricio y Mónica, a mi hermano Kevin y a mi abuelito Samuel, que todo este tiempo han estado detrás de mi apoyándome y dándome ánimo en los momentos más difíciles, llenándome de principios éticos y morales, y dándome un ejemplo de amor, perseverancia y fortaleza.

También les dedico a todos mis compañeros que durante todos estos años han venido compartiendo momentos y anécdotas que perdurarán durante toda la vida.

Finalmente, lo dedico a todos aquellos soñadores que día a día luchan por cumplir sus sueños a pesar de los diferentes obstáculos que se interpongan en su camino.

Christian



# CAPÍTULO 1: Introducción

## 1.1 Panorama general

Esta tesis está vinculada con la forma en el que las Tecnologías de la Información (TIC) pueden ayudar en la educación presencial para obtener un mejor rendimiento académico de los estudiantes. Dentro del contexto de este documento, se pretende diseñar e implementar un MOOC (Massive Open Online Course por sus siglas en inglés, en español son conocidos como Cursos en línea abiertos y masivos) para las tres primeras unidades de la asignatura de Algoritmos, Datos y Estructuras que se ofrece a los estudiantes de primer año en la Facultad de Ingeniería de la Universidad de Cuenca. Para esto se diseñó e implementó una plataforma tecnológica para dar soporte al despliegue del material educativo digital. Una vez implementada la plataforma MOOC, se la utilizó con los estudiantes matriculados en la asignatura mencionada.

Para el análisis de los datos se diseñará y construirá un modelo conceptual de datos, basado en estilos de aprendizaje y otras variables de relevancia, por medio del cual se podrá analizar y evaluar los resultados y la eficacia de la propuesta de enseñanza con MOOCs reuniendo evidencia empírica por medio de un caso de estudio.

## 1.2 Planteamiento del problema y justificación

En la actualidad, la educación de las masas se ha convertido en una problemática creciente puesto que, a nivel mundial, esta se ha visto incrementada en grandes porcentajes. Podemos abordar el caso práctico de México en relación al año 1945 en el cual se contaba con treinta mil estudiantes de licenciatura mientras que para el año 1985 había cerca de un millón (Rodríguez & Concheiro, 2013).

El sociólogo francés François Dubet se manifiesta con respecto a la educación de masas, sobre todo a los problemas de este sistema educativo, puesto que no cree de manera absoluta en la meritocracia (la escuela fija los lugares que van a ocupar los personajes en la sociedad a través



de sus méritos), ni está convencido de que la educación está hecha simplemente para reproducir los estratos sociales imperantes (a los ricos se les enseña a ser ricos y a los pobres a ser pobres).

La educación de masas está basada en la premisa de que si todos tienen acceso a la educación, ésta dará las herramientas para que cada individuo se vuelva lo que quiera ser. El problema es que los alumnos, al provenir de contextos desiguales, se enfrentan a la “libertad” que se les ofrece con distintos bagajes y experiencias personales. Los mejores alumnos son, con frecuencia, los más privilegiados socialmente y esto facilita que superen las pruebas escolares con éxito mientras que el resto tienen mayores dificultades para convertirse en estudiantes de manera plena (Dubet, 2011).

El aspecto social es solo una de las problemáticas actuales, por otra parte tenemos el aspecto de infraestructura y la falta de capacidad para enfrentar al número de bachilleres que año a año se postulan para las universidades, en el año 2011 – 2012 aproximadamente 170000 bachilleres se presentaron y aproximadamente 110000 lograron alcanzar un cupo para sus estudios (El Universo, 2011), con esto año a año se va incrementando el número de estudiantes que aspira un cupo, debido a la gratuidad de la educación pública y a como se están llevando a cabo los procesos para la selección de los postulantes a carreras universitarias.

Poner a disposición una serie de materiales digitales educativos utilizando plataformas tecnológicas puede favorecer los procesos de enseñanza y aprendizaje y a su vez dar soporte a los objetivos que persigue el Plan Nacional del Buen Vivir.

En la Universidad de Cuenca, cada semestre, se ofertan varias asignaturas que son comunes para los estudiantes de los primeros años de la Facultad de Ingeniería, de las que se puede nombrar a “Cálculo”, “Matemáticas Discretas”, “Algoritmos, Datos y Estructuras”, entre otras. Para el caso de la asignatura Algoritmos, Datos y Estructuras, se tiene semestralmente un promedio de 5 paralelos con 30 estudiantes cada uno. Contar con un número alto de estudiantes en el aula de clase dificulta al docente el seguimiento de las actividades teóricas y prácticas que se plantean en la asignatura. A esto se suma, la carga horaria semanal insuficiente (4 horas) en la que los docentes buscan cubrir un sílabo planificado previamente, sin dedicar horas exclusivas para prácticas que sirvan al estudiante como refuerzo a su aprendizaje. Esto se ha evidenciado en la tasa de aprobación de los estudiantes de la asignatura, según los datos históricos de años anteriores, donde la gran parte de los estudiantes reprueban la asignatura.

Cabe recalcar, que hoy en día, el entorno de aprendizaje, no se da únicamente en el aula de clase, sino que gracias a la inclusión de Tecnologías de la Información y Comunicación (TIC) se ha extendido, difiriendo el tiempo y el espacio en la que los estudiantes pueden acceder a los contenidos y apropiarse de estos de forma más provechosa, a su ritmo.

Un MOOC puede ser un complemento ideal de acompañamiento académico a las clases presenciales de la asignatura de *Algoritmos, Datos y Estructuras*, puesto que, permitiría al estudiante generar su propio conocimiento y responder inquietudes (mediante herramientas ofrecidas por el MOOC como los foros o las redes sociales) frente a temas que no pudieran ser abarcados por el docente en sus horas de clase. En este primer acercamiento del uso de los MOOCs, no se pretende quitar o eliminar la presencia del docente, sino más bien optimizar su tiempo y convertirlo en un guía para incentivar al estudiante a auto-educarse y crecer de una manera individual y auto-suficiente, proveyéndole *feedback* (retroalimentación) a medida que



se avanza en el curso. “La incorporación de los MOOCS como extensión del aula convencional favorece la labor del docente y en consecuencia mejora los procesos de enseñanza” (Toro & Robles, 2013).

A pesar que los MOOCs comparten algunas de las características de los cursos normales (una línea de tiempo predefinido o un cronograma de actividades), tienen características propias como la flexibilidad para abordar los contenidos sin un tiempo establecido y espacial (McAuley, Siemes, & Cormier, 2010).

Universidades en todo el mundo han comenzado a crear sus propias plataformas MOOCs; algunas inclusive uniéndose entre sí o con empresas privadas. Algunas de las plataformas más relevantes son (Yuan & Powell, 2013):

- Udacity
- Coursera: Universidad de Stanford (plataforma Course2go)
- Udemy
- Plataforma EdX: MIT, Harvard.
- Proyecto Course-builder: Google Inc.
- Miriada X: Telefónica, UniMOOC, Grupo Santander, y varias universidades españolas.
- Open Course Ware
- P2PU

Mediante el presente trabajo de tesis se pretende diseñar e implementar un MOOC sobre las tres primeras unidades de la asignatura Algoritmos, Datos y Estructuras, y ofrecerlo a los estudiantes de la asignatura. Con el propósito de analizar el rendimiento académico de los estudiantes, se desarrolló un modelo conceptual de datos que permitió abordar el estudio de los datos académicos de los estudiantes. Posterior a esto, se desarrolló un caso de estudio en el que se utilizó el modelo conceptual de datos para extraer conclusiones importantes que aportaron evidencia empírica sobre el impacto que tiene el utilizar un MOOC como apoyo a las clases presenciales.

El modelo de datos creados como producto del presente trabajo puede ser empleado para realizar futuros análisis, en los cuales se puedan implementar múltiples MOOCs en distintos campos de estudio y de esta manera poder evaluar los resultados y de esta manera poder tomar decisiones que puedan aportar en el rendimiento de los estudiantes.

## 1.3 Objetivos del trabajo

A continuación se describen los objetivos de la presente tesis.

### 1.3.1 Objetivo General

Evaluar el rendimiento académico de los estudiantes mediante un modelo conceptual de datos aplicado en un MOOC para la asignatura de Algoritmos, Datos y Estructuras.

### 1.3.2 Objetivos específicos

1. Introducir el concepto de MOOC bajo el contexto tecnológico y educativo.
2. Elegir una plataforma para MOOCs en base a un análisis comparativo de las plataformas disponibles en el mercado.



3. Desarrollar e implementar un MOOC para las tres primeras unidades de la asignatura de algoritmos, datos y estructuras.
4. Desarrollar un modelo conceptual que permita realizar el análisis de los datos obtenidos a partir del MOOC
5. Elaborar un caso de estudio en base a los resultados obtenidos con la puesta en marcha del MOOC.

## 1.4 Metodología de la investigación

Los pasos a seguir para el desarrollo de esta tesis son:

- **Búsqueda bibliográfica:** se buscan artículos científicos, revistas, libros y toda aquella fuente bibliográfica de la cual se pueda obtener información verificada concerniente al tema de investigación.
- **Análisis de plataformas:** con la información obtenida como base se procede a contrastar las características de las diferentes plataformas que permitirían cumplir con los objetivos del presente trabajo.
- **Síntesis y elección de plataforma:** con las ideas mucho más claras luego de comparar plataformas, el siguiente paso consistió en escoger la plataforma que más se adaptaría a las necesidades presentadas para la implementación del MOOC propuesto.
- **Diseño:** ya una vez seleccionada la herramienta más apropiada se emplearían las mejores prácticas para definir los distintos parámetros para la creación del MOOC.
- **Implementación:** en esta fase se creó el MOOC junto con todos los parámetros estipulados en el diseño.
- **Puesta en marcha:** consiste en llevar a cabo la ejecución del MOOC implementado previamente en un grupo de estudiantes determinado perteneciente a la Universidad de Cuenca.
- **Creación de modelo conceptual de datos:** se desarrolló a partir del análisis de los parámetros requeridos para evaluar el rendimiento académico de los estudiantes de la asignatura antes mencionada.
- **Caso de estudio:** en base al modelo conceptual de datos desarrollado se analizarán los datos obtenidos a partir del uso del MOOC por parte de los estudiantes.

## 1.5 Estructura de la tesis

En esta sección se presenta una breve descripción de los capítulos desarrollados en la presente tesis.

El **capítulo 2**, “Massive Open Online Courses (MOOC)”, presenta los conceptos generales sobre lo que es un MOOC, características, taxonomía, ventajas y desventajas. Se ha tomado en consideración los aportes realizados por diversos autores en los últimos 6 años para establecer el panorama general que desempeñará el término MOOC en la presente tesis.

El **capítulo 3**, “Plataformas para MOOC”, presenta un estudio de las plataformas MOOC más influyentes en la educación alrededor del mundo. Para cada una de ellas se realiza un análisis



de las características más relevantes, su arquitectura, ventajas y desventajas; además se realizó un diagrama de clases para modelar las funcionalidades que cada una de ellas ofrece. El capítulo concluye con un cuadro comparativo de las distintas plataformas analizadas, basado en la Norma ISO 25010:2010 para establecer la plataforma que mejor se ajusta a las necesidades para la implementación del MOOC de la presente tesis.

El **capítulo 4**, “Diseño e implementación de un MOOC”, detalla la metodología que se va a utilizar para la creación del MOOC. Además, se describen todos los pasos a seguir para realizar la implementación del mismo.

El **capítulo 5**, “Diseño e implementación de un Data warehouse para el análisis de resultados del MOOC”, presenta el diseño de un modelo conceptual que utilizará los datos recopilados por el MOOC en base al cual se implementará un Data warehouse que servirá de base para la creación del caso de estudio del siguiente capítulo.

El **capítulo 6**, “Caso de estudio”, presenta un caso de estudio basado en el modelo conceptual desarrollado en el capítulo 5 y que busca reunir evidencia empírica sobre los beneficios de integrar este tipo de plataformas a las clases presenciales dentro de una estrategia educativa

El **capítulo 7**, “Conclusiones y trabajos futuros”, se detallan las conclusiones que se han obtenido del desarrollo de la tesis y las líneas de trabajo futuro que se han generado con la misma.

**Anexo 1**, “Contenido y actividades MOOC”, describe el contenido que se va a presentar en el MOOC, los enlaces a las lecturas adicionales y videos utilizados.

**Anexo 2**, “Banco de preguntas para evaluaciones”, detalla las preguntas que se utilizarán en cada una de las evaluaciones del MOOC.

**Anexo 3**, “Herramientas para la implementación de un Data warehouse”, presenta una breve descripción de las herramientas tecnológicas utilizadas para el diseño e implementación de un data warehouse.

**Anexo 4**, “Herramientas tecnológicas utilizadas para el caso de estudio”, presentan la descripción de las herramientas tecnológicas utilizadas para el desarrollo del caso de estudio.

.



# CAPÍTULO 2:

## Massive Open Online Courses (MOOC)

### 2.1 Introducción

En este capítulo, el lector podrá encontrar una revisión del estado del arte acerca de los MOOCs. Primero, se define y describe el término MOOC, así también se da una mirada a los antecedentes que dieron origen a los MOOC. Segundo, se da una mirada a las bondades y beneficios que un MOOC puede ofrecer. Tercero, se muestra las distintas características de un MOOC tanto del lado tecnológico como del pedagógico. Cuarto, se hace una clasificación de los MOOCs en función de la teoría de aprendizaje teniendo como principales a los xMOOC y a los cMOOC, también se sugiere un método alternativo de clasificación mediante el tipo de plataforma que acoge al MOOC que puede tener licenciamiento abierto o cerrado. Después, se realiza una descripción de las plataformas más importantes para desplegar MOOCs que existen en la actualidad. Finalmente, se presenta desde la perspectiva general de las universidades, los docentes y los alumnos de todo el mundo, las distintas ventajas y desventajas que puede tener un MOOC.

### 2.2 ¿Qué es un MOOC?

MOOC es el acrónimo en inglés de *Massive Open Online Course*, nombre dado por Dave Cormier de la Universidad de la Isla del Príncipe Eduardo en el 2008 (Creed-Dikeogu & Clark, 2013). La traducción de MOOC al español es *Curso en Línea Masivo y Abierto*, a continuación se repasa el significado del acrónimo MOOC:

- Es un **Curso** debido a que el contenido que lo conforma está estructurado y secuenciado, de tal manera que tiene un inicio y un final (Siemens, 2013).



- Está **en Línea** ya que todo el acceso a los materiales e interacciones que se realizan son llevados en entornos virtuales utilizando Internet como canal de comunicación.
- Es **Abierto** por el hecho de que generalmente son de libre acceso sin la necesidad de pagar una matrícula ni de prerrequisitos.
- Es **Masivo** por la cantidad de estudiantes que pueden involucrarse en el curso, dicha cantidad muchas veces puede fácilmente alcanzar los varios miles; en algunas ocasiones la extensa cantidad de alumnos ha permitido formar sub-redes en las cuáles se podían compartir varias similitudes como el lenguaje, ubicación geográfica y otras características que facilitaban el desarrollo del curso (Siemens, 2013).

Los conceptos anteriores son de común acuerdo y pocos autores han tratado de redefinir el termino MOOC.

Por otro lado, Bartolomé (2014) en su obra *MOOC: 4+2 años de expectativas y resultados*, señala que los primeros cursos masivos fueron de tipo radiofónicos ofertados en el año 1922 por la Universidad de Nueva York, estos cursos eran abiertos porque no se requería de una suscripción pagada, ni título universitario, ni alguna especialización para poder acceder a estos cursos, y eran masivos porque estaban orientados a una gran cantidad de estudiantes; los estudiantes empleaban el correo ordinario para matricularse y posteriormente para enviar las tareas. Esta iniciativa fue luego adoptada por otras universidades como Columbia, Harvard, Kansas State, Ohio State, New York University y Purdue.

Los MOOCS como hoy se los conoce surgieron en el año 2008 (Smith & Eng, 2013), cuando Stephen Downes y George Siemens desarrollaron un curso en línea denominado Conectivismo (Kop & Hill, 2008) y Conocimiento Conectivo (Tschofen & Mackness, 2012). Sin darse cuenta crearon una tendencia que se puso rápidamente en movimiento. Los dos profesores de la universidad canadiense de Manitoba introdujeron el que sería un paso gigantesco en la educación: el curso abierto masivo en línea (MOOC). Estos profesores construyeron un MOOC usando la teoría de la conectividad que consiste en que "el conocimiento se distribuye a través de una red de conexiones, por lo tanto que el aprendizaje consiste en la capacidad de construir y atravesar esas redes" (Scardilli, 2013), refiriéndose a la red de conexiones como grupos de estudiantes en donde el conocimiento se encuentra distribuido en cada uno de ellos.

El modelo empleado por los MOOCs en la actualidad es básicamente el mismo que fue empleado por los cursos radiofónicas de 1922, pudiéndose mencionar como las diferencias más notables el uso de la multimedia en los MOOCs y el Internet como canal de transmisión.

## 2.3 ¿Para qué sirve un MOOC?

Entre los principales objetivos de un MOOC están el poder reunir un número amplio de alumnos (pudiendo llegar a los miles), materiales de curso y un medio de transferencia de información; todo esto limitado solo por las capacidades tecnológicas y el costo de las mismas (El Ahrache, Badir, Tabaa, & Medouri, 2013). Los MOOCs por lo general no requieren haber tomado otros cursos previamente y permiten al estudiante el hecho de poder elegir entre una gran variedad de cursos que inclusivamente pueden ser dictados por especialistas en cualquier campo de estudio (Creed-Dikeogu & Clark, 2013).

Los MOOCs son considerados como una alternativa para mediar los contenidos que se encuentran disponibles en la web y la guía por parte de un docente hacia el estudiante

(Siemens, 2013). Aunque la propuesta de los MOOC está orientada a la educación a distancia, también podría emplearse como una herramienta de soporte a las clases presenciales, así como también a las que llevan la modalidad de semi-presenciales, proporcionando a los estudiantes acceso a materiales o recursos extras que permiten que los estudiantes puedan mejorar o incrementar sus conocimientos (Bonvillian & Singer, 2013).

Son herramientas como los teléfonos inteligentes, internet de alta velocidad, Wi-Fi y la computación en la nube que han permitido el desarrollo de los MOOC y han creado un entorno virtual de aprendizaje que rompe las barreras físicas y geográficas (El Ahrache, Badir, Tabaa, & Medouri, 2013). Es decir, los MOOCs pueden ser tomados por los estudiantes sin importar el lugar del planeta en el que estos se encuentren ni una ubicación geográfica específica para todo el curso, simplemente pueden hacerlo desde sus hogares, oficinas u otra ubicación en la que tengan acceso a una computadora o dispositivo con Internet.

Aunque los MOOCs pueden tener fechas establecidas para el seguimiento de las lecciones, realización de actividades o la rendición de evaluaciones, no existe una restricción de horarios específicos o más bien existe una flexibilidad de horarios (Creed-Dikeogu & Clark, 2013). Por ejemplo, una evaluación puede estar disponible solo durante ciertos días por lo cual el estudiante podrá realizar esa evaluación a cualquier hora en la que este vea conveniente, siempre y cuando el horario escogido por el estudiante esté dentro de las fechas establecidas por el MOOC.

Al mismo tiempo, en algunos casos la información generada por los MOOCs y el seguimiento de los mismos es recogida y almacenada para su interpretación o posterior análisis, entre estos casos está edX que emplea esta información para conocer y mejorar el proceso de aprendizaje de los estudiantes a través de la tecnología (El Ahrache, Badir, Tabaa, & Medouri, 2013). En estos casos cuando la información del MOOC es recogida y analizada se ven involucrados expertos en enseñanza, los cuáles poco a poco irán descubriendo la manera apropiada en la que se pueda introducir el uso de los MOOCs en la educación tradicional, de tal manera que de a poco se llegue a alcanzar una metodología de educación ideal, en la cual se tome lo mejor de los MOOCs y de la educación tradicional (Bonvillian & Singer, 2013).

Finalmente, otro beneficio que tiene un MOOC es el poder otorgar certificados. Es decir, hay MOOCs en los que una vez terminado y aprobado el curso se paga una cuota, a cambio la institución gestora del MOOC entrega un certificado indicando la aprobación del MOOC. Este tipo de certificados de poco a poco van ganando su espacio, a tal punto que algunas empresas los validan para poder entregar salarios diferenciados a sus empleados. Por otra parte, debido a que muchas de las instituciones creadoras de MOOCs son Universidades, algunas de ellas están ofreciendo créditos de grado a cambio de completar uno de estos cursos (Bonvillian & Singer, 2013).

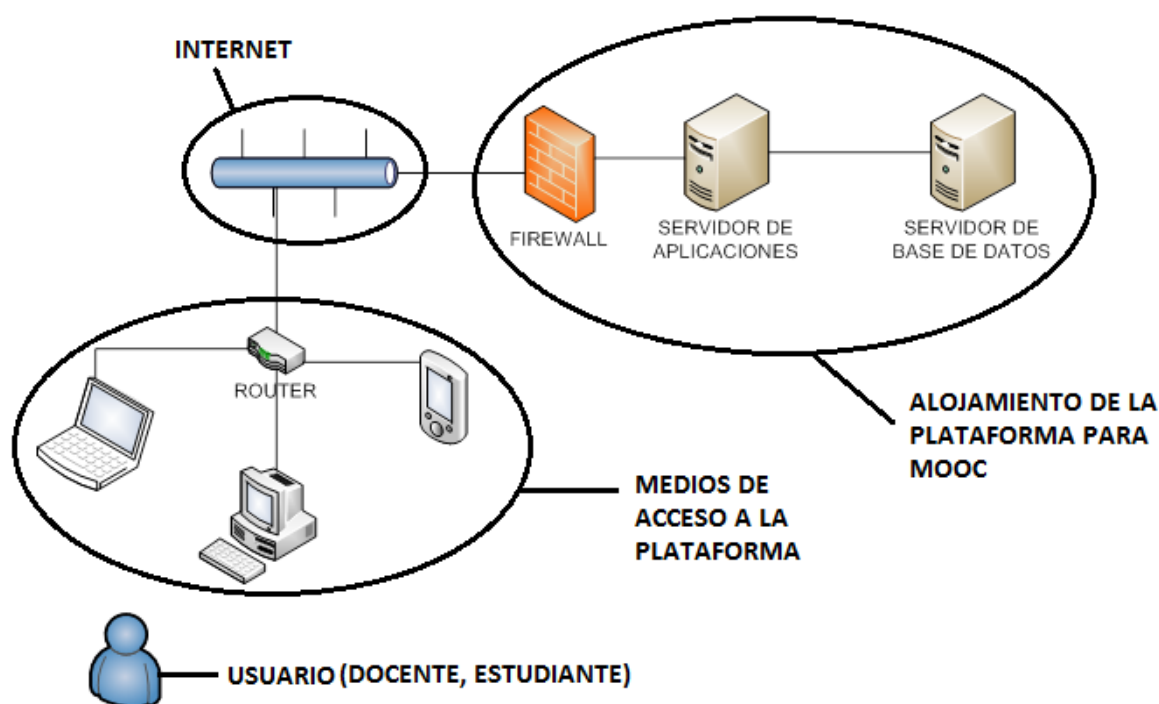
## 2.4 Características de un MOOC

Dado que en un MOOC se usa los medios tecnológicos a los que hoy en día se tiene acceso para expandir la educación sobre sus limitantes naturales (ubicación geográfica, cantidad de alumnos a los que se puede llegar), es conveniente fragmentar las características en un ámbito tecnológico y un ámbito educativo.



Del lado tecnológico, un MOOC requiere de una plataforma informática sobre la que se despliegan los contenidos, esta plataforma es una aplicación web<sup>9</sup> que debe desplegarse sobre un servidor de aplicaciones<sup>10</sup>, este servidor es el encargado de manejar la disponibilidad y gestionar las distintas conexiones que puedan generarse simultáneamente. La plataforma está ligada a una base de datos la cual guarda la información de los distintos cursos así como también los datos de los diferentes usuarios que se encuentren registrados. Dentro del equipo de desarrollo de los MOOCs se incluye un equipo técnico que se encarga de dar soporte tecnológico a las distintas personas involucradas (Marauri Martínez de Rituerto, 2014).

En la siguiente *figura 2.1* se muestra la infraestructura necesaria para que la plataforma pueda estar disponible para los usuarios.



*Figura 2.1: Infraestructura tecnológica para un MOOC.  
Fuente: Elaboración propia.*

Del lado educativo los MOOCs de igual forma que un curso virtual o una clase presencial proponen objetivos de aprendizaje, actividades, retroalimentación y evaluaciones. Los MOOCs pueden incluir diversos tipos de materiales como un sílabo, lecturas, tareas, lecciones, pruebas, actividades de aprendizaje que pueden ser interactivas y evaluaciones al final del curso. Las clases en un MOOC, de manera habitual contienen un conjunto de videos de una duración de 5 a 10 minutos cada uno.

Por lo general, un MOOC tiene una duración de 2 a 6 semanas aproximadamente y en casos excepcionales pueden alcanzar hasta las 8 semanas, además de ser accesibles las 24 horas del día, los 7 días de la semana. La mayoría de los contenidos son entregados asincrónicamente, es

<sup>9</sup> **Aplicación web:** es un conjunto de herramientas disponibles en un servidor web a las que un usuario puede acceder a través de Internet o de una intranet mediante un navegador.

<sup>10</sup> **Servidor de aplicaciones:** es un tipo de servidor que se encarga de desplegar aplicaciones web, estas aplicaciones responden a solicitudes que llegan a través de la red desde equipos cliente remotos o desde otras aplicaciones.





decir, que los estudiantes pueden acceder a los mismos de forma diferida en tiempo y espacio. Sin embargo, algunos MOOCs también pueden tener contenidos simultáneos como sesiones interactivas, en las cuales se requiere que los participantes accedan al MOOC a una hora y fecha específica.

Todo el curso está diseñado por un equipo de docentes y en algunos casos existe la colaboración de especialistas en ciertas temáticas, se podría mencionar el caso en el cual el curso trate temáticas acerca de Android en donde la colaboración vendría por parte de empresas como Google o empresas encargadas del desarrollo de aplicaciones para Android; también se podría indicar el caso en el que la temática es el análisis de datos en donde la colaboración vendría de empresas como Facebook si los datos serían provenientes de redes sociales o empresas como Amazon si los datos provienen de fuentes como el comercio electrónico (Billings, 2014).

En cuanto a lo que se refiere a los elementos principales que conforman un MOOC, Kop (citada por El Ahrache, Badir, Tabaa, & Medouri, 2013) menciona a los siguientes: instructores, estudiantes, tema, material y contexto.

- Instructores o docentes: son los encargados de llevar el proceso de aprendizaje brindando el material adecuado, iniciando la comunicación entre los estudiantes y gestionando las distintas evaluaciones en función de los objetivos de aprendizaje propuestos.
- Estudiantes: son todos aquellos interesados en el tema del MOOC, los estudiantes pueden o no estar detrás de la obtención de un certificado formal.
- Tema: es obtenido en función a los estudiantes que se pretende llegar, al material que se presentará, al contexto en el que se desarrollará y más que todo al criterio de los instructores. El objetivo que se pretende alcanzar con la creación del tema es el de generar énfasis y de proveer una cobertura extensa.
- Material: puede estar distribuido en diferentes sitios pudiendo accederse a través de distintas maneras como por ejemplo: documentos en Google Drive o Dropbox, páginas web. El material puede ser de distintos tipos (videos, documentos, presentaciones, gráficos, animaciones, entre otros).
- Contexto: está constituido por los diferentes actores que pueden conformar el entorno de aprendizaje. Entre dichos actores pueden estar: redes sociales, soluciones tecnológicas, información de origen convencional, variados tipos de esquemas de transferencia de información, sistemas de comunicación, objetivos de aprendizaje y los grupos que constituyen la oferta de cada curso.

De igual manera se podría añadir a los elementos principales de un MOOC los siguientes:

- Evaluaciones y actividades: generalmente son llevadas a cabo por medio de cuestionarios con preguntas de opción múltiple o también a través de revisión por pares<sup>11</sup>; aunque también puede existir actividades que se realicen en grupos, en donde se emplean las herramientas sociales para la interacción de los estudiantes.
- Herramientas sociales: medios en los que los usuarios pueden compartir ideas o inquietudes, estas inquietudes pueden ser respondidas tanto por docentes como por otros estudiantes; entre este tipo de herramientas podemos tener: foros, wikis, blogs, facebook, twitter, skype.

---

<sup>11</sup> Revisión por pares: metodología de validación en la que se intercambian documentos escritos o archivos entre miembros de igual rango de conocimientos.

A continuación, en la figura 2.2 se pueden ver los elementos que conforman un MOOC:

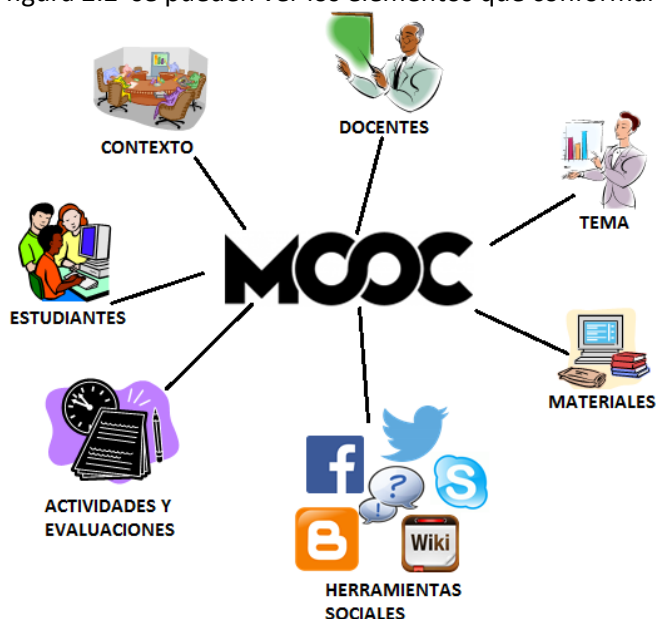


Figura 2.2: Elementos de un MOOC.

Fuente: Elaboración propia.

Un MOOC tiene marcadas diferencias con aquellos cursos que se llevan a cabo de forma presencial, entre estas diferencias Marauri Martínez de Rituerto (2014) menciona las siguientes:

- Los grupos de trabajo en un MOOC pueden ir variando en las diferentes fases del curso: creación, desarrollo, diseño instruccional y puesta en ejecución.
- Los alumnos involucrados en un MOOC no están restringidos a una ubicación geográfica.
- La atención personalizada que se le puede ofrecer a un alumno en un curso tradicional es casi inexistente en un MOOC.
- Los sistemas de evaluación son quizás el punto en el que exista una mayor diferencia. En un MOOC no se puede realizar un seguimiento personalizado de los alumnos y además para evitar la carga a los docentes de tener que revisar grandes cantidades de materiales se emplean métodos como la autoevaluación o la evaluación por pares.

En la *Tabla 2.1* se realiza una comparación entre las diferentes actividades que pueden ser encontradas en un MOOC y en una clase presencial:

	MOOC	CLASE TRADICIONAL
<b>Actividad</b>	<ul style="list-style-type: none"> <li>○ Debates, a través del uso de foros.</li> <li>○ Trabajos prácticos, solo pueden ser documentados mediante el uso de archivos digitales; la generación de objetos físicos es muy complicado de llevar a cabo.</li> <li>○ Resolución de problemas, solo se puede usar archivos digitales para el desarrollo.</li> <li>○ Trabajo escrito, solo se</li> </ul>	<ul style="list-style-type: none"> <li>○ Debates, generados dentro del salón de clase.</li> <li>○ Trabajos prácticos, puede tener como fin la generación de un objeto físico como una maqueta, o también se puede tener una descripción en papel.</li> <li>○ Resolución de problemas, las soluciones pueden ser descritas en papel o también se puede usar archivos digitales.</li> <li>○ Búsqueda de información, la</li> </ul>

	<p>puede emplear archivos digitales.</p> <ul style="list-style-type: none"> <li>○ Trabajos grupales, todas las actividades mencionadas pueden ser llevadas a cabo a través de grupos de estudiantes, pero en este caso se emplean las herramientas sociales como medios de interacción.</li> <li>○ Cuestionarios, que por lo general contienen preguntas de opción múltiple.</li> </ul>	<p>información puede ser empleada para generar debates o espacios de discusión.</p> <ul style="list-style-type: none"> <li>○ Juegos de simulación o interpretación de roles.</li> <li>○ Trabajo escrito, se puede utilizar el papel para la descripción al igual que archivos digitales.</li> <li>○ Trabajo grupal, todas las actividades mencionadas pueden ser llevadas a cabo a través de grupos de estudiantes, en los cuáles puede existir interacción física entre los distintos miembros.</li> </ul>
<b>Evaluaciones</b>	<ul style="list-style-type: none"> <li>○ Cuestionarios, están conformados en su gran mayoría por preguntas de opción múltiple.</li> <li>○ Escritas, se puede describir la solución a problemas propuestos en archivos digitales como documentos de word o excel.</li> </ul>	<p>Escritas, la soluciones son escritas y entregadas en papel al maestro.</p> <p>Orales, el maestro realiza preguntas al estudiante y el estudiante tiene que responder mediante el diálogo.</p>
<b>Metodologías de calificación</b>	<ul style="list-style-type: none"> <li>○ Automática, el sistema se encarga de realizar la calificación.</li> <li>○ Revisión por pares.</li> </ul>	<ul style="list-style-type: none"> <li>○ Manual, el maestro es quién se encarga de la calificación.</li> <li>○ Revisión por pares.</li> </ul>

Tabla 2.1: Comparación de Actividades de un MOOC y de una clase tradicional.

Fuente: Elaboración propia.

## 2.5 Taxonomía de MOOC

Kop, Fournier y Hill (citados por Creed-Dikeogu & Clark, 2013) describen cuatro tipos de actividades que comparten todos los tipos de MOOCs, estos son: agregación, donde se da disponibilidad de recursos como lecturas, videos y otros recursos web en el curso; re-mezcla, en donde se vuelven a emplear los distintos recursos del curso pero de otra manera, esta puede ser en un blog o en foros de discusión; re-utilización, en donde se fomenta a los estudiantes a crear algo por su cuenta a través de los conocimientos adquiridos; y la alimentación posterior, en donde se promueve a que los estudiantes compartan su trabajo. Sin embargo, lo que separa a los MOOCs es la teoría de aprendizaje que estos emplean, por un lado están los cMOOCs que están basados en el conectivismo (Grünwald, Meinel, Totschnig, & Willems, 2013); y por el otro lado tenemos a los xMOOCs que se orientan más hacia la teoría conductista<sup>12</sup> con unos pequeños tintes constructivistas<sup>13</sup> (Marauri Martínez de Rituerto, 2014).

Los cMOOCs están centrados en los intereses y necesidades personales de formación de los estudiantes por lo tanto las pruebas y las evaluaciones objetivas toman un segundo plano; los

<sup>12</sup> **Conductismo:** es la interacción históricamente construida entre el individuo y su ambiente físico, biológico y social.

<sup>13</sup> **Constructivismo:** postula la necesidad de entregar al alumno herramientas que le permitan construir sus propios procedimientos para resolver una situación problemática.



docentes más bien se encargan de brindar el contenido inicial del curso fomentando al estudiante a que sea el quien lo complemente (Méndez García, 2013). En cuanto a lo que se refiere a la comunicación se emplean variadas maneras tales como: sistemas de mensajería instantánea, redes sociales, blogs, wikis, correo electrónico, entre otros (Marauri Martínez de Rituerto, 2014).

Por lo que se refiere a los xMOOCs, estos tomaron su nombre de la plataforma edX, la cual fue fundada por el Instituto Tecnológico de Massachusetts y la Universidad de Harvard (Grünewald, Meinel, Totschnig, & Willems, 2013). El modelo pedagógico empleado por este tipo de MOOCs es en gran parte más tradicionalista en el cuál se percibe al docente como el que posee todo el conocimiento y al estudiante como el consumidor de dicho conocimiento (Siemens, 2013). En este tipo de cursos el equipo docente se encarga de elaborar los contenidos y de planificar el desarrollo del curso en su totalidad, es por esto que las instituciones implicadas manejan parámetros mínimos de calidad debido a que son estas las que serán beneficiadas o criticadas por las opiniones referentes a los cursos ofertados (Marauri Martínez de Rituerto, 2014).

Schoenack (2013) se refiere a la falta de interés que puede originar un cMOOC debido a bajos factores motivacionales, como el hecho de que no exista la suficiente confianza en los alumnos para formar parte de la red de conocimiento. Mientras que los xMOOC tienen la limitante de llevar los cursos de una forma parecida a las clases magistrales, es decir, un maestro transmite el conocimiento para todos los alumnos. Es por eso que, Schoenack propone un tercer tipo: los mesoMOOCs; este modelo prototipo lo que sugiere es la combinación de los dos ya existentes, uniendo la escalabilidad brindada por los xMOOCs y el conectivismo ofrecido por los cMOOCs, mientras que también se podrían emplear prácticas efectivas de aprendizaje en línea y la andragogía<sup>14</sup>.

Así mismo Siemens (2013) se refiere a otro tipo, los quasi-MOOCs los cuales básicamente no son cursos, pero están compuestos por tutoriales basados en la Web en forma de recursos educativos abiertos<sup>15</sup>, algunos recursos de este tipo que se podrían mencionar son los ofrecidos en Khan Academy y el OpenCourseWare del MIT, el objetivo de estos materiales es el de brindar la ayuda en el aprendizaje de tareas específicas como por ejemplo la resolución de un problema matemático, estos recursos pueden tener un nexo entre sí pero no se encuentran empaquetados como un solo curso.

Otro tipo de MOOC que ha venido apareciendo recientemente en el campo de los MOOCs es el pMOOC (project-based MOOC) o también conocido como task-based MOOC (Holotescu, Cretu, & Grosseck, 2013). Este tipo de MOOC emplea una mezcla de las teorías Constructivista e Instruccional<sup>16</sup>, en la que los estudiantes están guiados por un individuo o por materiales grabados, para poder proseguir en el curso se deben ir realizando distintas tareas. Se promueve el trabajo en grupo, aunque las tareas pueden ser realizadas de forma individual (Three types of MOOC, 2013). En este tipo de MOOC lo que se pretende es que el estudiante adquiera ciertas aptitudes y destrezas a través del desarrollo de actividades (Peláez López). En

<sup>14</sup> **Andragogía:** es el conjunto de técnicas y metodologías orientadas a la educación de personas adultas.

<sup>15</sup> **Recursos Educativos Abiertos:** son documentos o material multimedia con fines relacionados a la educación, cuya principal característica es que son de acceso libre.

<sup>16</sup> **Instructivismo:** el conocimiento existe independientemente del alumno, y se transfiere al estudiante por el maestro.

la figura 2.3 se muestran los diferentes tipos de MOOC y sus principales características en base a la teoría de aprendizaje empleada.

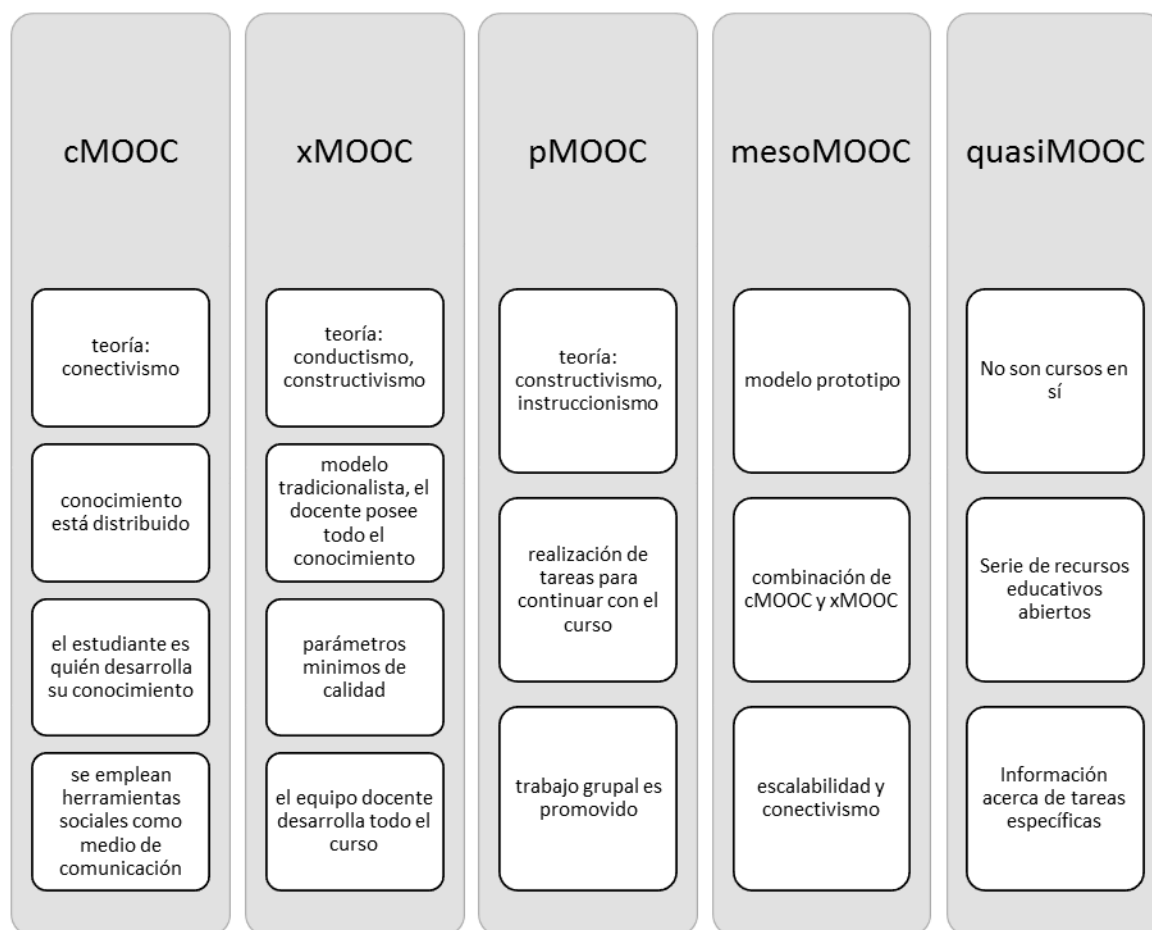


Figura 2.3: Tipos de MOOC según la teoría de aprendizaje.  
Fuente: Elaboración propia.

Como una forma alterna de clasificación se podría dividir a los MOOCs según la plataforma sobre la que estos se despliegan, se puede decir que están las plataformas open-source y las privativas o cerradas. Por el lado de las plataformas open-source estas tienen liberado su código en la web, permitiendo que personas modifiquen este código y creen su propia plataforma o realicen cambios que consideren necesarios; en este tipo de plataformas podríamos mencionar a edX bajo licencia AGPL<sup>17</sup> y Google Course Builder bajo Apache License<sup>18</sup> 2.0, un ejemplo de plataforma creada a partir del código liberado por Google Course Builder es UniMOOC. Por el otro lado están las plataformas privativas o cerradas que solo brindan sus soluciones a través del internet pero no permiten ver el código fuente mucho menos alterarlo, entre este tipo de plataformas están Udemy, Udacity, Coursera, entre otras. En la figura 2.4 se puede observar las principales características de esta forma alternativa de clasificación.

<sup>17</sup> **AGPL (Affero General Public License):** es una licencia copyleft derivada de la Licencia Pública General de GNU diseñada específicamente para asegurar la cooperación con la comunidad en el caso de software que corra en servidores de red.

<sup>18</sup> **Apache License:** es una licencia de software libre creada por la Apache Software Foundation (ASF). requiere la conservación del aviso de copyright y el disclaimer, pero no es una licencia copyleft, ya que no requiere la redistribución del código fuente cuando se distribuyen versiones modificadas.

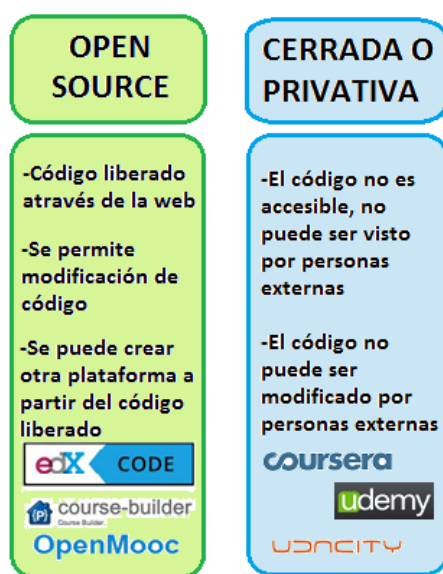


Figura 2.4: Tipos de MOOC según el tipo de plataforma.  
Fuente: Elaboración propia.

## 2.6 Plataformas MOOC

Los MOOCs nacieron de la necesidad por parte de instituciones de educación superior de poder difundir de forma masiva contenidos, dando así acceso libre a la información y el conocimiento, no solo a estudiantes sino a cualquier persona que tenga interés en un tema. Es por esto que son varias las universidades que se han unido alrededor del mundo y han lanzado sus propias plataformas sobre las que se implementan los MOOCs.

Así también, la empresa privada se ha visto involucrada en este tipo de alianzas junto con las universidades, existiendo alianzas que tienen fines de lucro por detrás (Coursera, Udemy, Udacity) mientras que otras no (EdX, Google Course Builder, P2PU). A continuación se realiza un análisis de algunas de las más importantes plataformas disponibles en la actualidad.

1. **EdX:** es una plataforma sin fines de lucro que fue lanzada bajo la alianza de la Universidad de Harvard y el Instituto Tecnológico de Massachusetts con un presupuesto aproximado de \$60 millones (Yuan & Powell, 2013). Al momento se ofertan más de 200 cursos en diferentes áreas como ciencia, física, computación, matemáticas, entre otras; dichos cursos son impartidos no solo por las instituciones fundadoras de la plataforma sino también por universidades de alto prestigio y empresas asociadas provenientes de todo el mundo (Edx, 2014). La plataforma es de código abierto, razón por la cual instituciones externas pueden impartir sus propios cursos. La plataforma está desarrollada en Python, JavaScript, Ruby y Node.js, posee todo un conjunto de herramientas que permiten la gestión del curso y también el análisis de datos producidos por el curso. Entre las grandes ventajas que presenta esta plataforma está la constante evolución tecnológica, y también la gran adaptabilidad a diferentes tipos de pantalla.



2. **Coursera:** es una organización con fines de lucro creada bajo una inversión inicial de \$22 millones contribuidos por New Enterprise Associates y Kleiner, Perkins, Caufield & Byers Education (Yuan & Powell, 2013). Fue fundada por Andrew Ng<sup>19</sup> y Daphne Koller, profesores de la Universidad de Stanford. Hoy en la actualidad se ofertan más de 600 cursos repartidos en diferentes áreas de especialidad e idioma, así también como de diferentes instituciones alrededor del mundo gracias a convenios realizados. Por otra parte, a pesar de que la plataforma no es open source emplea muchas herramientas de este tipo para su desarrollo como: Scala, Django, Nginx, Ubuntu, Cassandra y Node.js, pero en un futuro la organización planea lanzar sus proyectos bajo licencias de código abierto (Coursera, 2014).
3. **Udacity:** es una organización con fines de lucro creada por Sebastián Thrun, David Stavens and Mike Sokolsky, esta contó con una inversión inicial de \$21.1 millones provenientes de las firmas de capital de riesgo Charles River Ventures and Andreessen Horowitz (Yuan & Powell, 2013). En la actualidad se ofrecen más de 30 cursos de los cuáles la gran mayoría tratan temáticas tecnológicas, estos cursos son creados en colaboración con grandes empresas como Google, Facebook, Nvidia, Autodesk, entre otras. La plataforma se desarrolla con tecnologías como: Python, MongoDB, PostgreSQL, y Javascript (Udacity, 2014).
4. **Udemy:** otra organización con fines de lucro fue fundada en el 2010 por la unión de varias empresas inversoras como Insight Venture Partners, Lightbank, MHS Capital y otras 500 start-ups, la inversión inicial fue de \$16 millones (Yuan & Powell, 2013). Udemy ofrece herramientas a sus usuarios para poder crear sus propios cursos, después estos cursos son promovidos por la plataforma y permiten a sus creadores obtener una ganancia a partir del cobro de ingreso a los estudiantes. Actualmente se ofrecen más de 18000 cursos repartidos en varias categorías y en una gran variedad de idiomas, la plataforma cuenta con más de tres millones de usuarios. La plataforma se desarrolla bajo tecnologías como: Python, PHP, Django, MySQL, CentOS, Jenkins y Github. (Udemy, 2014).
5. **MiríadaX:** apareció a finales del 2012 en España como una de las primeras plataformas separadas del entorno Norteamericano, contexto común en el que se desarrollaban gran parte de las plataformas ya existentes. Para llevar a cabo el desarrollo de MiríadaX se unieron varias entidades como Telefónica Learning Services, Banco Santander, Fundación CSEV y Universia, el objetivo principal de este proyecto era el poder ofrecer MOOCs impartidos por universidades pertenecientes a la red Universia (Medina-Salguero & Aguaded, 2014). Al día 25 de Septiembre de 2014, Miríada X tenía 943.390 usuarios inscritos en sus cursos, número presentado en la página web propia de la plataforma.
6. **Course Builder:** es un proyecto de código abierto lanzado por Google en el 2012. Todo comenzó después de que el departamento *Research at Google* lanzara un MOOC llamado Power Searching with Google, el cual llegó a tener la admirable cantidad de 155.000 estudiantes registrados. Dado tal suceso la empresa Google decidió liberar la tecnología empleada para la creación de dicho MOOC bajo el nombre de Course Builder.

---

<sup>19</sup> Andrew Ng es un profesor asociado en el departamento de Ciencias de la Computación y del departamento de Ingeniería Electrónica por cortesía en la Universidad de Stanford.





Este proyecto brinda las herramientas para que cualquier usuario pueda crear su propio curso y difundirlo sin necesidad de un hosting (Norvig, 2012).

De igual manera existen otras plataformas que han venido ganando su espacio en el mundo de los MOOCs, entre estas tenemos:

- o **Future Learn:** fue fundada en diciembre del 2012 como resultado de la unión de Open University y otras 11 universidades del Reino Unido, la iniciativa era poder crear una plataforma local propia como respuesta al creciente mercado de los MOOCs (Shaw, 2012).
- o **Khan Academy:** es del tipo de los quasiMOOCs, fue creada en el 2006 por Salman Khan como una iniciativa sin fines de lucro, en la cual se pudiera compartir material educacional de manera libre, esta plataforma registra más de diez millones de estudiantes por mes accediendo a más de seis mil videos instruccionales y a más de cien mil problemas prácticos en áreas como las matemáticas, física, química, economía, informática, entre otras. Esta iniciativa ha recibido el apoyo de fundaciones pertenecientes a personas como Bill Gates y Carlos Slim, también de empresas como Google (Khan Academy, 2014).
- o **P2PU (Peer to Peer University):** es una organización sin fines de lucro creada en el año 2009 con el apoyo de las fundaciones Hewlett y Shuttleworth, su iniciativa es el crear cursos en los cuales se reúna el material proveniente de distintas fuentes y se entregue a los estudiantes, para que sean estos estudiantes los cuales mediante interacciones directas entre ellos vayan generando su propio conocimiento (Hafner, 2010).
- o **CourseSites:** es una plataforma lanzada por la empresa Blackboard, esta plataforma permite que cualquier docente pueda crear su propio curso de manera gratuita, la cantidad de cursos que un mismo docente puede crear no tiene un límite, pero se le permite al docente tener solo cinco cursos disponibles al mismo tiempo con un límite de 500MB para la carga de contenidos tanto de parte del docente como del estudiante (COURSEsites, 2014).
- o **Canvas Network:** apareció en noviembre del 2012 como parte del deseo de la empresa Instructure de lanzarse al mercado de los MOOCs, Canvas network permite crear a los docentes sus cursos sin ningún costo, así también permite a los estudiantes seguir cualquier curso de forma gratuita (Hill, 2012).
- o **Open MOOC:** es una plataforma de tipo open source lanzada bajo la licencia Apache 2.0, apareció en octubre del 2012 con el soporte de la UNED (Universidad Nacional de Educación a Distancia) de España, el proyecto Open MOOC fue empleado por la propia UNED para crear su propia plataforma UNEDCOMA (Wukman, 2012).

Día a día son más las instituciones que se deciden optar por este tipo de tecnologías para ofrecer mejores opciones, ya sea a sus estudiantes o a personas externas que no tienen ninguna relación con dichas instituciones.

A continuación, en la *figura 2.5* se muestra como ha sido la evolución de las plataformas mencionadas.



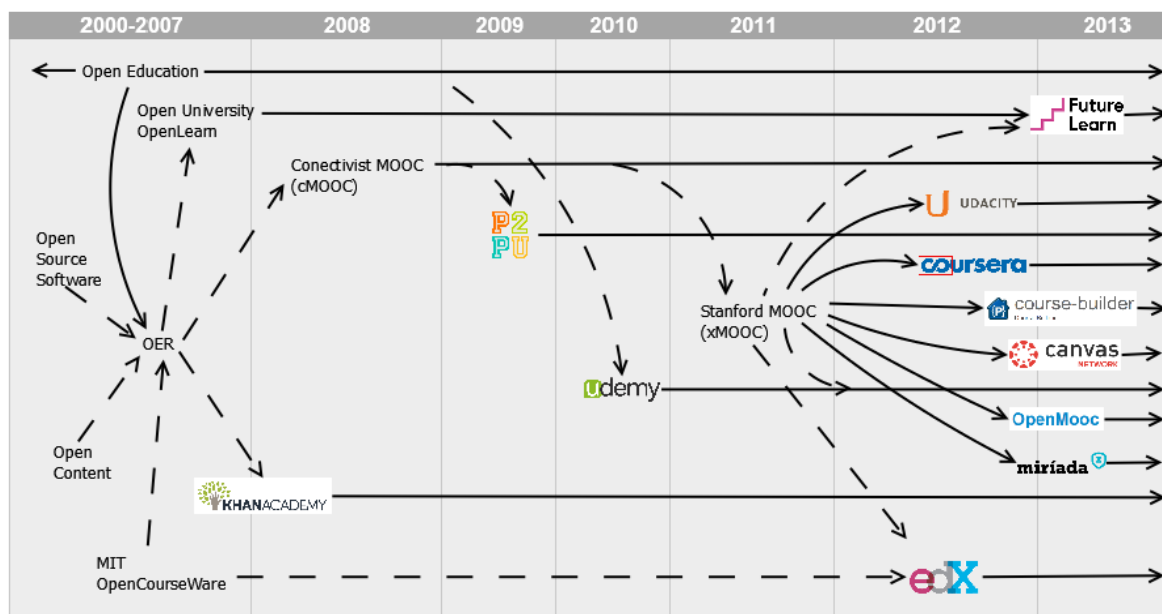


Figura 2.5: Línea de Tiempo de los MOOCs.

Fuente: Elaboración propia.

## 2.7 Ventajas y desventajas de los MOOC

Las ventajas y desventajas de un MOOC serán analizadas desde tres perspectivas diferentes: las universidades, los docentes y los estudiantes.

### UNIVERSIDADES

#### Ventajas

- Los MOOCs permiten que las universidades puedan compartir el conocimiento generado dentro de sus establecimientos, dando acceso a la educación a personas de todas las partes del mundo (Méndez García, 2013).
- La aprobación de los MOOC pueden ser ofrecidos como créditos de aprobación para las carreras, reduciéndose la necesidad de espacio físico y dándose disponibilidad para otras necesidades (Bonvillian & Singer, 2013).
- La información generada por los MOOCs puede ser procesada y analizada para realizar estudios sobre el impacto de la tecnología en los procesos de aprendizaje y desarrollar analítica de datos que permitan tomar acciones proactivas en la enseñanza (El Ahrache, Badir, Tabaa, & Medouri, 2013).

#### Desventajas

- Los costos que puede acarrear la implementación de una infraestructura tecnológica necesaria para desplegar una plataforma para MOOCs pueden llegar a ser elevados, a tal punto que no todas las universidades del planeta pueden cubrir estos costos (Billings, 2014).

- Por otro lado la producción de los materiales o recursos necesarios para cada MOOC también pueden tener costos elevados y el hecho es que las universidades no lanzan un solo curso MOOC sino varios, necesitando elevados presupuestos para poder poner en marcha estos cursos (Billings, 2014).
- La cantidad de alumnos que se inscriban en un MOOC puede no llegar a cumplir las expectativas iniciales, por lo que se podría recurrir a gastos excesivos innecesarios (Marauri Martínez de Rituerto, 2014).

## DOCENTES

### Ventajas

- Los docentes pueden emplear los MOOCs como una forma de promocionar su trabajo en el campo de la enseñanza, de tal manera que mientras mayor sea su éxito dentro de los MOOCs mayor será el reconocimiento que los docentes tengan, a tal punto que estos podrían ser invitados frecuentemente a congresos y conferencias internacionales (Marauri Martínez de Rituerto, 2014).
- La calificación de las actividades y evaluaciones puede realizarse de forma automática por parte del sistema o también se emplea la evaluación por pares, de esta manera se evita la sobrecarga de trabajo que puede tener un docente al tener que realizar la calificación de forma manual (Marauri Martínez de Rituerto, 2014).

### Desventajas

- La copia y el plagio no pueden ser controlados en las diferentes actividades y evaluaciones que se puedan presentar en un MOOC (Siemens, 2013).
- Un problema que se podría presentarse es al momento de generar las evaluaciones, debido a que los docentes en gran parte emplean la calificación automática, reduciendo las evaluaciones a usar solo preguntas de opción múltiple (Creed-Dikeogu & Clark, 2013).
- Si un docente es novato en la generación de contenidos para el MOOC puede tener serios problemas, en especial si se considera que no habrá una relación directa profesor-alumno y que la diversidad de alumnos que se pueda tener en un MOOC puede conllevar a la falta de conocimientos previos acerca de la temática del curso, el bajo conocimiento del uso de la tecnología e inclusive en muchos casos el idioma en el cual se impartirá el MOOC (Méndez García, 2013).
- Un problema de gran amplitud se presentaría si para crear un MOOC habría que considerar culturas o contenidos minoritarios, debido a qué se necesitaría mucho más trabajo e investigación para poder generar actividades y programas de formación (Bartolomé, 2014).
- Si los docentes desearan que el MOOC estuviera disponible en varios idiomas, esto implicaría crear materiales distintos para cada idioma pudiendo presentarse inclusive la necesidad de crear otro MOOC completo.



## ESTUDIANTES

### Ventajas

- Dado que un MOOC es abierto se reducen las limitaciones de acceso a la educación, permitiéndose a los estudiantes tener acceso a los materiales generados por docentes y expertos que en muchos de los casos provienen de universidades de alto prestigio como Harvard, el MIT o Stanford (Billings, 2014).
- Los MOOCs dan soporte a una gran cantidad de personas que viven en países en vías de desarrollo, con bajos ingresos económicos, con discapacidades, que tienen la necesidad de adquirir habilidades para sus trabajos, quieren ser emprendedores o simplemente personas que desean expandir sus conocimientos (Welsh & Dragusin, 2013).
- Los estudiantes pueden acceder a los contenidos de un MOOC desde cualquier lugar a la hora que ellos crean conveniente, de esta manera no se necesita tener una ubicación geográfica específica ni tampoco un horario fijo durante el curso (El Ahrache, Badir, Tabaa, & Medouri, 2013).

### Desventajas

- El porcentaje de estudiantes que completan un curso es demasiado bajo, esto se debe a que existen casos en donde la motivación no es intrínseca como en el caso de los cursos tradicionales; es decir, por lo general los MOOCs no representan créditos de grado o no existe el pago de una inscripción previa (Siemens, 2013).
- El idioma puede ser uno de los mayores inconvenientes de los MOOCs debido a que un estudiante podría estar interesado en seguir un MOOC que no estaría disponible en un idioma entendible para el estudiante.

## 2.8 Recapitulación

Los MOOCs son cursos abiertos y masivos en línea que pueden ser tomados por cualquier persona que tenga interés en alguna temática específica sin la necesidad de pagar una suscripción previa; estos cursos están disponibles a través de plataformas especializadas en el Internet. Entre los principales objetivos que se persiguen a través de un MOOC están el poder reunir un número amplio de alumnos (fácilmente podrían llegar a los miles), materiales de curso y un medio de transferencia de información; los MOOCs dan a las personas el poder elegir hasta cursos que podrían ser impartidos por profesores y expertos provenientes de universidades u organizaciones de cualquier parte del mundo, todo esto con solo poseer un dispositivo con una conexión a Internet. Además, algunos de los MOOCs al ser completados dan la oportunidad de poder obtener un certificado con el ajuste de una cierta cuota, mientras que algunas universidades ofrecen algunos créditos de grado por tomar estos cursos.

La clasificación de los MOOCs está dada básicamente por la teoría de aprendizaje que estos emplean, por un lado tenemos los cMOOCs que emplean el conectivismo, en este tipo se promueve la idea del aprendizaje como un proceso creativo y social de conexión de nodos de conocimiento; por el otro lado tenemos a los xMOOCs que emplean la teoría conductista con una ligera tendencia hacia el constructivismo, en este tipo el equipo docente es el encargado de elaborar la planificación y los contenidos del curso en su totalidad.



En la actualidad se puede encontrar una gran variedad de plataformas que ofrecen MOOCs, entre las más importantes tenemos: edX que es la plataforma creada por la Universidad de Harvard y el MIT, Coursera, Udacity, Udeemy que es una de las plataformas con mayor cantidad de cursos ofertados, MiríadaX que es una propuesta española de la red Universia y la empresa privada, y finalmente Google Course Builder como un proyecto de código abierto que inclusive permite ser modificado para crear una nueva plataforma.



# CAPÍTULO 3:

## Plataformas para MOOC

### 3.1 Introducción

Antes del año 2008, acceder al contenido de cursos en prestigiosas universidades del mundo era muy complicado para la mayoría de las personas, siendo uno de los principales inconvenientes el costo económico que representaba matricularse en cada uno de ellos.

Hoy en día, gracias a los MOOCs, varias universidades de renombre a nivel mundial, permiten a una gran cantidad de personas acceder a cursos de forma gratuita y sin que el horario y la distancia sea un impedimento.

Por lo general, quien toma un MOOC no se necesita conocimientos previos sobre un tema específico para acceder al mismo, aparte de tener acceso a un computador con conexión a internet. La mayor parte del tiempo, la instrucción educacional o académica del estudiante no es un impedimento para matricularse en un MOOC.

Las plataformas MOOC existentes son las que se listan a continuación:

- Google Course Builder
- EdX
- Coursera
- MiriadaX
- OpenCourseWare
- Udacity
- Eliademy
- Open Class
- Open MOOC

## 3.2 Plataformas para MOOC

- Google Course Builder



Figura 3.1: Mapa Geográfico de los países en donde se utiliza Google Course Builder.

Fuente: <http://www.ovtt.org/2013-11-11/unimooc-plataforma-mooc-de-cursos-abiertos-para-emprendedores>

Como se puede observar en la figura 3.1, Google Course Builder tiene un nivel de aceptación mayor al 25% con respecto a otras plataformas (Delgado Kloos, 2014) pese a que fue lanzado recientemente (2013); esta plataforma tiene un mayor nivel de incidencia en Centro y Sudamérica al igual que en Europa como en algunos países de África y Asia. Para acceder a esta plataforma se debe ingresar al siguiente enlace <https://code.google.com/p/course-builder/>

- EdX



Figura 3.2: Mapa Geográfico de los países en donde se utiliza EdX.

Fuente: <http://ralstonconsulting.com/blog/>

EdX fue la primera plataforma en poner a disposición el contenido de distintos cursos, es por esta razón que el nivel de aceptación a escala mundial ha tenido una gran aceptación a nivel mundial y eso lo demuestra la figura 3.2; sin embargo estas plataformas son desarrolladas de preferencia en idioma inglés, esto no resultaría un inconveniente de no ser por la escasa aceptación que los cursos de habla hispana han tenido. Esto genera un inconveniente al considerarlo para el desarrollo pero, por



motivos de comparación resulta interesante analizar los puntos que lo han convertido en una herramienta de un nivel de aceptación elevado. Para acceder a esta plataforma se debe acceder <https://www.edx.org>

- Coursera



Figura 3.3: Mapa Geográfico de los países en donde se utiliza Coursera.

Fuente: <http://www.dunsurfin.com/my-mooc-life-so-far-part-2/coursera-student-map/>

Como se puede ver en la figura 3.3, Coursera al igual que EdX ha tenido una gran acogida dado su carácter de pionera, el problema de esta plataforma consiste en que es una herramienta que exige formar parte del portal, no es de código abierto y se depende en extremo del sitio web principal. Los cursos de esta plataforma están disponibles en <https://www.coursera.org>

- Miríada X



Figura 3.4: Mapa Geográfico de los países en donde se utiliza Miríada X.

Fuente: <http://nurseteando.com/wp-content/uploads/2013/04/Miriadax.png>

Miríada X es la primera plataforma creada íntegramente para Iberoamérica, y evidentemente como muestra la gráfica, es en donde ha tenido más acogida. El mayor inconveniente que posee es que esta no puede manejar ningún nivel de independencia sino restringe a que la universidad se registre y esté bajo la tutela del portal general. La plataforma está disponible en <https://www.miriadax.net/>

- Open Course Ware

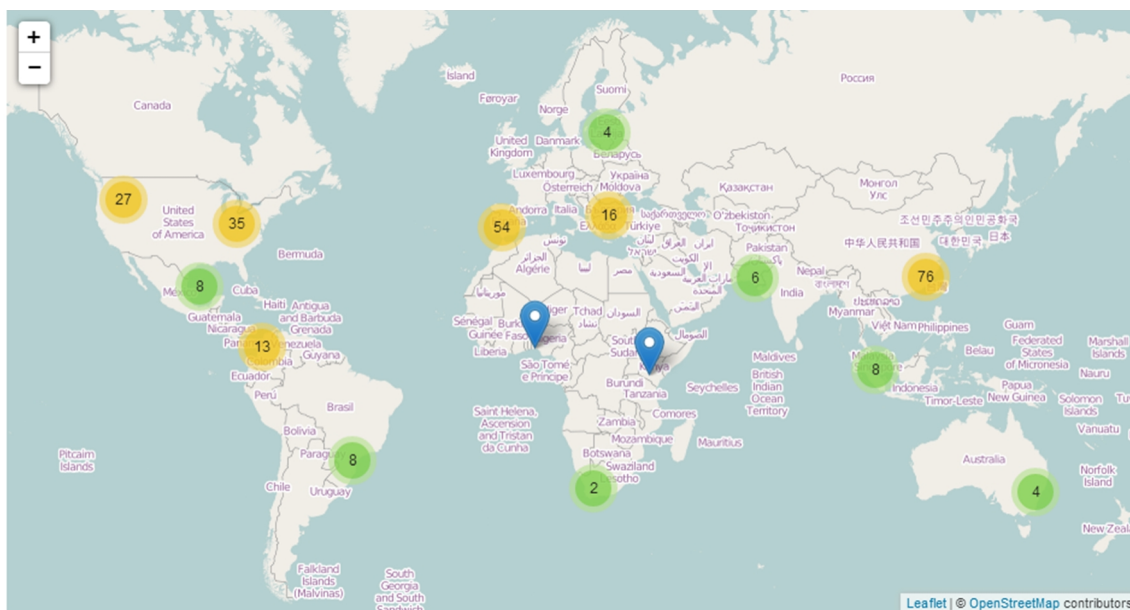


Figura 3.5: Mapa Geográfico de los países en donde se utiliza Open Course Ware.

Fuente: <http://www.oecconsortium.org/members/>

Open Course Ware requiere que la Universidad que desee publicar sus MOOCs en su plataforma esté registrada en el consorcio que rige la herramienta, por otra parte, cómo se puede observar en la figura 3.5, Open Course Ware, no ha tenido suficiente acogida mundial, sin embargo se debe reconocer que ha llegado a alcanzar a una mayor cantidad de países en África y Oceanía.

- Udacity



Figura 3.6: Mapa Geográfico de los países en donde se utiliza Udacity

Fuente: <http://www.meetup.com/Udacity/>

Udacity es una organización educativa con ánimo de lucro, fundada por Sebastián Thrun, ex docente de la Universidad de Stanford. La plataforma requiere que el estudiante domine el idioma inglés, ya que los cursos son exclusivamente en este idioma. Por otro



lado, si una Universidad desea publicar sus cursos en esta plataforma debe solicitar unirse a la plataforma como colaborador, y bajo todas las políticas que maneje la plataforma. Para acceder a esta plataforma se debe ingresar al siguiente enlace <https://www.udacity.com>

- Eliademy



Figura 3.7: Mapa Geográfico de los países en donde se utiliza Eliademy  
Fuente: <http://www.eurodl.org/?p=archives&year=2012&halfyear=1&article=507>

Eliademy es una plataforma para MOOC desarrollada inicialmente para las universidades de Finlandia. Es multilinguaje y ha sido adoptado incluso por empresas que desean dar capacitaciones a sus empleados. La plataforma requiere validar los cursos que se publiquen, y sólo permite que los usuarios que sean miembros de Eliademy sean los creadores de los cursos. Para acceder a esta plataforma se debe ingresar al siguiente enlace <https://www.eliademy.com>

- Open Class



Figura 3.8: Mapa Geográfico de los países en donde se utiliza Open Class  
Fuente: <http://consiliera.wordpress.com/>

Open Class es un entorno de aprendizaje gratuito en la nube creado por la editorial Pearson. Está en su versión beta y entre sus principales características están el ambiente de aprendizaje dinámico, escalable y totalmente basado en la nube, que estimula el

aprendizaje social y permite la distribución de contenidos a escala masiva para los estudiantes.

- Open MOOC



Figura 3.9: Mapa Geográfico de los países en donde se utiliza OpenMOOC.

Fuente: [//www.ligonier.org/blog/ligonier-connect-worlds-largest-christian-mooc](http://www.ligonier.org/blog/ligonier-connect-worlds-largest-christian-mooc)

Open MOOC es una solución de software libre para la creación y gestión de MOOC, es impulsado por la Universidad Nacional de Educación a Distancia de España (UNED), Telefónica y Universia. Entre las funcionalidades de la plataforma se destacan las actividades de autoevaluación y herramientas que permiten valorar los progresos en el aprendizaje. Está aún en su versión beta.

La tabla 3.1, muestra la incidencia de cada plataforma MOOC a nivel mundial:

Plataforma	Número Países
<b>Coursera</b>	190
<b>EdX</b>	160
<b>Google Course Builder</b>	140
<b>Open Course Ware</b>	30
<b>Miriada X</b>	23
<b>Udacity</b>	20
<b>Eliademy</b>	20
<b>Open Class</b>	14
<b>Open MOOC</b>	13

Tabla 3.1: Incidencias de empleo de plataformas MOOC en el mundo.

Tabla del autor.

### Porcentaje del uso de plataformas MOOC alrededor del mundo

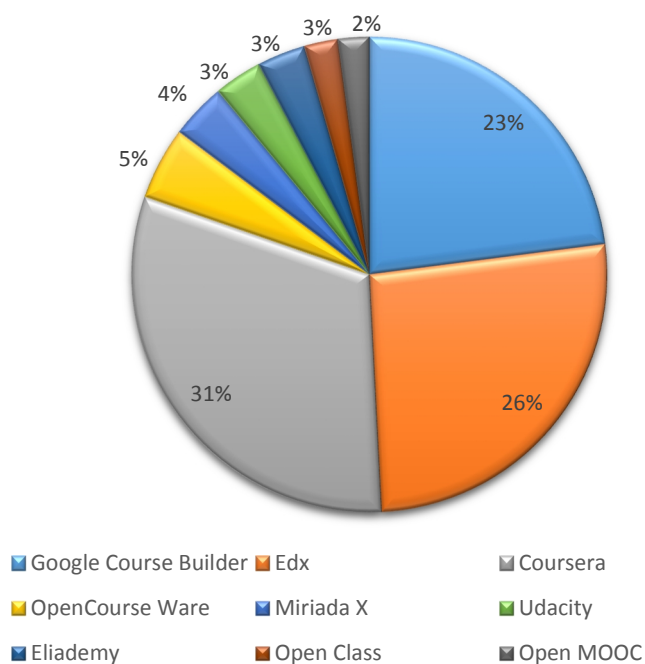


Figura 3.10: Porcentaje de incidencias de plataformas MOOC en el mundo.

Fuente: Elaboración Propia.

Como se muestra en la figura 3.10, Coursera es la plataforma más utilizada alrededor del mundo, seguida muy de cerca por EdX y Google Course Builder.

La tabla 3.2 muestra la madurez de las plataformas tecnológicas a lo largo del tiempo:

Plataforma	Tiempo de madurez en el mercado
Udacity	Desde el año 2011. Meses: 43
Coursera	Desde el año 2011. Meses: 42
EdX	Desde el año 2011. Meses: 39
OpenCourse Ware	Desde el año 2012. Meses: 35
Open Class	Desde el año 2012. Meses: 30
Open MOOC	Desde el año 2012. Meses: 25
Miriada X	Desde el año 2013. Meses: 20
Eliademy	Desde el año 2013. Meses: 18
Google Course Builder	Desde el año 2013. Meses: 15

Tabla 3.2: Tiempo en el mercado de las plataformas MOOC.

Tabla del autor

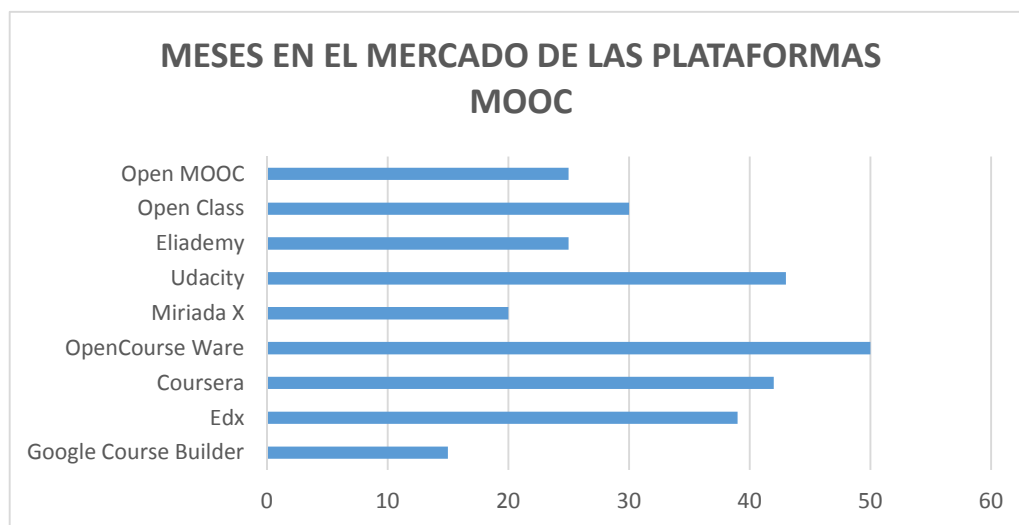


Figura 3.11: Tiempo en el mercado de las plataformas MOOC.

Fuente: Elaboración Propia.

En la figura 3.11, se puede observar que Open Course Ware, es la plataforma que más tiempo tiene en el mercado, seguida por Udacity, Coursera y EdX. Además, se puede observar que Google Course Builder junto con MiriadaX son las plataformas que menos tiempo llevan en el mercado.

En base a estos gráficos estadísticos, se van a comparar para esta tesis las siguientes plataformas ya que son las más utilizadas a nivel mundial considerando el tiempo que las mismas llevan en el mercado:

- Google Course Builder
- EdX
- Coursera
- MiriadaX
- OpenCourseWare

### 3.2.1 Google Course Builder

*Google Course Builder* es una plataforma de educación en línea de código abierto (Apache<sup>9</sup> 2.0). Se usa para crear un curso en línea sea esta para una materia de una determinada carrera universitaria, capacitaciones profesionales, o un producto determinado de una empresa. *Google Course Builder* posee las siguientes características (Google, 2014):

- Posee un nivel de escalabilidad<sup>10</sup> amplio puesto que se ejecuta con la infraestructura de Google.
- Es de código abierto.



Figura 3.12: Logotipo Course Builder.

Fuente: <http://techcetera.co/google-course-builder-o-construya-su-propio-curso/>

<sup>9</sup> Apache es el servidor web más utilizado a nivel mundial. Es un proyecto de código abierto y uso gratuito, multiplataforma, muy robustos y que destaca por su seguridad y rendimiento (Digital Learning, 2012).

<sup>10</sup> Escalabilidad describe la facilidad con la que se pueden agregar o quitar componentes de un sistema a la vez que se mantiene la confiabilidad de éste (Microsoft, 2007).

- Soporta múltiples cursos en la misma instancia (multiusuario).
- Soporta *Google Analytics* y *Google Tag Manager*.
- No requiere de programación para crear o ejecutar un curso.
- La implementación es rápida.

*Google Course Builder* fue creado en el año 2013, como un experimento de Google para montar un MOOC denominado *Power Searching with Google*. El curso tuvo una gran acogida a nivel mundial, y esto sumado al auge de los MOOC, el uso de esta herramienta se ha extendido tanto en su desarrollo por parte de Google como de otras entidades educativas (Google, 2014).

Una de las principales razones para el éxito de esta plataforma, es que se encuentra alojado en *Google App Engine*<sup>11</sup>, la nube de Google, por lo que los usuarios no deben preocuparse por alquilar o comprar servidores para alojar el MOOC, montar bases de datos, etc., puesto que *App Engine* se encarga de gestionar todos estos parámetros automáticamente, es decir, en caso de no contar con un dominio propio, *Google App Engine* proporciona uno con la siguiente estructura: *midominio.appspot.com*, donde “*midominio*”, es el identificador del curso en la web.

Otra de las principales características de *Google Course Builder*, es su facilidad para probar nuevos contenidos sin necesidad de subirlo a la nube, para lo cual se necesita únicamente tener instalado Python<sup>12</sup> y el *App Engine SDK*<sup>13</sup>.

### Arquitectura de un MOOC en Course Builder

*Google App Engine* ejecuta eficientemente aplicaciones escalables por medio de una entidad denominada *Balanceador de Carga*, el mismo que se encarga de asignar los recursos (servidores) a cada una de ellas.

En la figura 3.13, se puede observar la arquitectura de *Google Course Builder*.

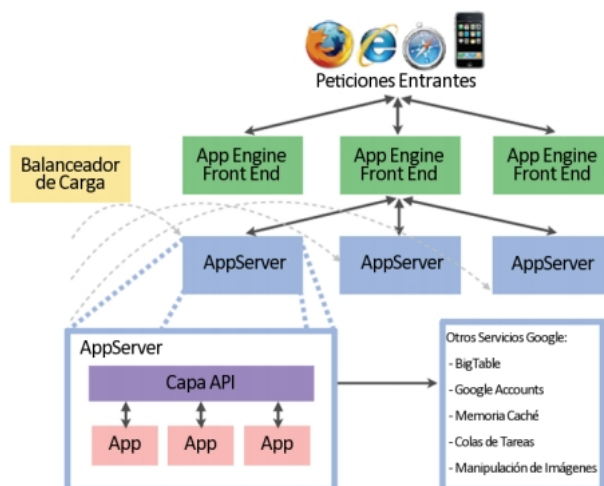


Figura 3.13: Arquitectura de *Google Course Builder*.

Fuente: (Sánchez Cabañas, 2011)

<sup>11</sup> Google App Engine es un servicio de alojamiento web que presta Google de forma gratuita que permite ejecutar aplicaciones sobre la infraestructura de Google (Wikipedia, 2014).

<sup>12</sup> Python es un lenguaje de programación interpretado, que soporta programación orientada a objetos, programación imperativa y programación funcional. Constituye el lenguaje de programación con el cuál fue creado *Google Course Builder* (Wikipedia, 2014).

<sup>13</sup> AppEngine SDK es un software de Google que permite administrar las aplicaciones creadas localmente (Google, 2014).

El funcionamiento de Course Builder es el siguiente: los usuarios acceden al MOOC, el mismo que es implementado por un número de servidores establecido por el balanceador de carga, en función de los recursos necesarios y/o disponibles. Por su parte, los servidores utilizan la API<sup>14</sup> que les proporciona Google y al mismo tiempo se puede hacer uso de otros servicios de la infraestructura de Google como la BigTable<sup>15</sup> o las cuentas de usuario (Google Accounts) (Sánchez Cabañas, 2011).

### Roles que intervienen en Course Builder

Actualmente en Course Builder existen dos roles que son el autor del curso y el administrador. Estos roles permiten separar la implementación técnica del diseño del curso. Esta división ayuda a cada persona o equipo encargado a centrarse en competencias respectivas. Esta división puede ser creada a discreción de la organización dado que puede darse la posibilidad de que sea solo una persona la encargada de cumplir los dos roles.

### 3.2.2 COURSERA



Figura 3.14: Logotipo de Coursera.

Fuente: <https://www.coursera.org/>

Coursera es una plataforma educativa asociada con las universidades y organizaciones más renombradas de todo el mundo como la Universidad de California, Universidad de Edimburgo, Universidad de Melbourne, Universidad de Hong Kong, Universidad de Virginia, etc. Por medio de esta colaboración, se ofrece cursos gratuitos en línea accesibles a todas las personas por medio de un computador e Internet (Coursera, 2014).

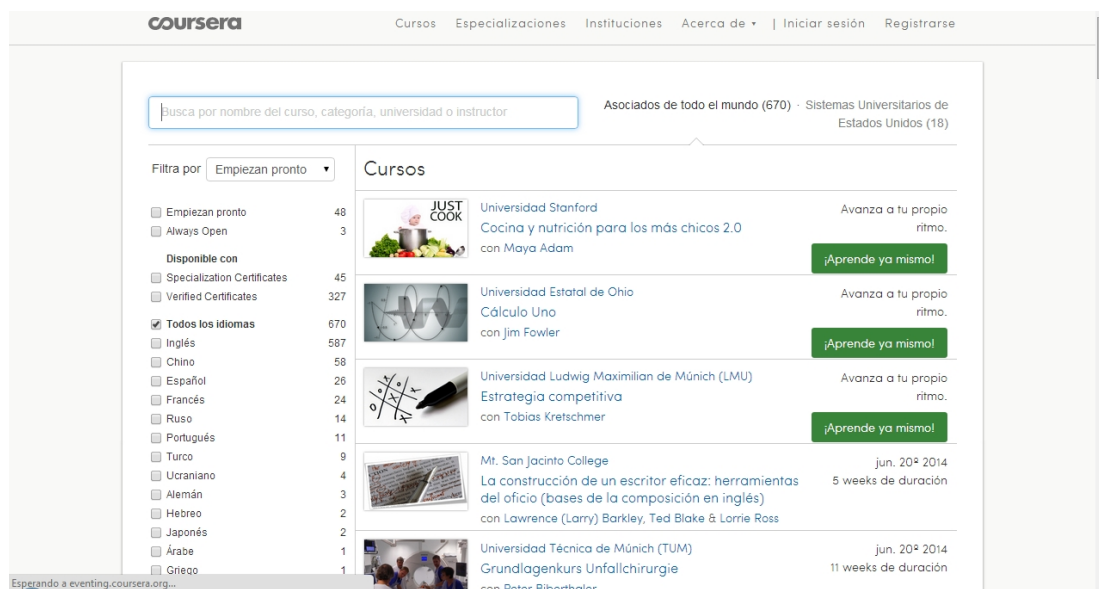


Figura 3.15: Página principal de los cursos ofertados en Coursera.

Fuente: (Coursera, 2014)

<sup>14</sup> API, Interfaz de programación de aplicaciones: Representa el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción (Wikipedia, 2014).

<sup>15</sup> Big Table es un sistema de gestión de base de datos creado por Google con las características de ser distribuido, de alta eficiencia y propietario (Wikipedia, 2013).



Coursera está basado en un entorno virtual de enseñanza y aprendizaje adaptado donde el perfil de usuario cuenta con una descripción y la posibilidad de añadir enlace a redes sociales. No permite acceder a la lista de participantes. Los foros se pueden dividir en sub-foros y los mensajes se pueden votar (positivo o negativo), de cada hilo se muestran el número de votos, de post y de visitas y se marcan las intervenciones del profesor. En el foro cuando se pulsa sobre el nombre aparece la descripción del usuario y un enlace al perfil completo, además de toda la actividad en el foro.

En algunos cursos cuenta con un enlace a un wiki externo (beta) en el que pide registro. Se intuye que tiene un wiki interno pues en algunos enlaces en la url aparece “wiki” pero ni se plantea como tal ni se puede editar por parte de los estudiantes.

Las herramientas que ofrece Coursera para dar apoyo a los procesos de aprendizaje son (Matas, 2012):

- **Calendario del curso:** Aquí se describe la planificación del curso semana a semana. Se indica el objetivo del tema, qué lecturas y vídeos hay que estudiar, si hay algún test o ensayo que realizar, etc.
- **Información del curso:** Explicación sobre el funcionamiento del curso.
- **Video Lecturas:** Los vídeos grabados por el profesor con las explicaciones correspondientes a cada tema. Cada video suele durar una media de 15 minutos y cada tema contiene varios.
- **Pruebas:** Test de conocimientos.
- **Asignación de ensayos:** Lugar para entregar y leer los ensayos.
- **Foros de discusión:** Foros de discusión y encuentro entre estudiantes (pueden llegar a ser 70000 o más en cada curso), profesor y equipo de apoyo. Se suelen formar grupos de estudio por idiomas, países o intereses. Por ejemplo, grupos de español o de compañeros de aula.
- **Preguntas frecuentes:** Las preguntas frecuentes. En este apartado hay información importante, como los libros que conviene leer o si se entregará, o no, un certificado de finalización del curso.
- **Unirse a una reunión:** Para encuentros de estudiantes en todo el mundo.
- **Subtítulos:** El curso proporciona subtítulos en inglés para ayudar a la comprensión de los vídeos, pero algunos alumnos de forma altruista añaden los subtítulos para otros idiomas. Los diferentes apartados se adaptan a la forma de trabajo del profesor, pero todos los cursos tienen una estructura similar.
- **Interfaz sencilla**

Todas las opciones del curso se encuentran a la vista, lo que denota una gran usabilidad en el entorno del curso.

- En el cuerpo central, se encuentra el feed de noticias, que corresponden a las novedades del curso, los contenidos que también se pueden leer en los mails que te van enviando.
- En la columna de la derecha se pueden ver los temas del curso en sí, un pequeño estado de las tareas vigentes, y a continuación el temario de las vídeos lecturas que se ha ido subiendo y que están disponibles para visualizar.

#### Estructura del curso (Valverde, 2012):

**Syllabus** – Es un directorio con todos los enlaces y material que se citan o se muestran en los vídeos tutoriales. Es una sección donde el maestro coloca las lecturas recomendadas a manera de videos de conferencias que permiten aumentar la gama de conocimientos.

**Formato de evaluación continua** - A través de tres tipos de modalidades básicas: cuestionarios sobre las vídeo lecturas, tareas de redacción y un examen final. Los cuestionarios se pueden realizar dos veces (en una segunda instancia se puede llegar a alcanzar únicamente el 75% del total de puntos).

**Áreas de seguimiento y comunicación:** Existe un foro, una wiki con material en google docs, posibilidad de editar los subtítulos de los vídeos, etc. Y finalmente un meetup para que te unas a la comunidad de Coursera en tu ciudad.

**Certificaciones:** Coursera ofrece certificaciones a los participantes de los cursos que hayan completado el régimen de estudios.



Figura 3.16: Página de Coursera para obtener certificados de MOOC aprobados.

Fuente: (Coursera, 2014)



### 3.2.3 Miríada X

Miríada X es un proyecto de formación en línea que tiene su origen a principios del año 2013 financiado por el Banco Santander y Telefónica, a través de la Red Universia y Telefónica Learning Services y basado en la plataforma de software libre WEMOOC. Ofrece cursos en línea masivos y en abierto (Massive Open Online Courses, MOOCs) de forma gratuita y aptos para cualquier usuario interesado en el contenido del mismo.



Su éxito la ha llevado a convertirse en una plataforma de formación online de referencia no solo a nivel español sino también europeo, en el que más de un 35% de los MOOCs provienen de universidades españolas según Open Education Europa, siendo Miríada X factor clave en la evolución educativa española según muestra el informe de la Sociedad de la información en España en el año 2013. Al día 29 de abril de 2014, Miríada X tenía 649.378 usuarios inscritos en sus cursos (Colaboradores de Wikipedia, 2014).

Figura 3.17: Logotipo de MiriadaX.  
Fuente: <https://www.miriadax.net>

MiriadaX presenta las siguientes características.

- Directorio de cursos y universidades.
- Entorno de creación de cursos autogestionado por el docente.
- Herramientas para el seguimiento y la gestión de cursos, incluyendo *learning analytics*.
- Herramientas de interacción entre los estudiantes (social learning).
- Uso de Mozilla Open Badges<sup>16</sup> para acreditación y motivación (Octeto, 2012).

MiriadaX está basada en el aprendizaje colaborativo y el uso libre tanto de los recursos educativos como de las tecnologías innovadoras. La plataforma lleva casi dos años en actividad (20 meses) y cuenta con más de 750000 alumnos inscritos de los cuales 100000 ya han superado los cursos (Universia, 2014).

#### Elementos de la plataforma Miríada X

- **Cursos**

En el espacio dedicado a cursos nos encontramos, situado en la parte izquierda, un buscador general de cursos y otro por temática, además de una relación de áreas de conocimiento que nos permite filtrar cursos por categorías. Y en la parte derecha se sitúa el catálogo de cursos que se han ofertados, los que están en desarrollo actualmente y los que están por determinar sus fechas para futuras convocatorias.

En cuanto refiere a la jerarquía visual de la información de cursos, esta se encuentra estructurada de tal forma que se compone de una breve descripción sobre las personas a las que va dirigido el curso y el objetivo que se plantean, la fecha de inicio, la situación de la inscripción, duración, universidad que lo imparte, idioma, temática y enlace sobre la ficha técnica del curso.

<sup>16</sup> Mozilla Open Badges es una nueva iniciativa de Mozilla y la fundación MacArthur, con el cual se pretende mostrar las habilidades adquiridas a modo de logros o insignias, también conocida como Badges. En MiriadaX, se podrá exportar las insignias conseguidas con la realización de los cursos dentro de la plataforma a Mozilla Open Badges para desde ahí poder compartirlas en redes sociales, portales de empleo, webs, etc.

Los cursos de MiriadaX están abiertos a todo el público, independientemente de su nivel educativo, sin necesidad de pertenecer a alguna universidad específica. Los estudiantes pueden inscribirse en los cursos ofertados de forma fácil y gratuita, con el único requisito de querer ampliar sus conocimientos (Universia, 2014).

- **Universidades**

En el espacio dedicado a las universidades se muestran las diversas instituciones de educación superior que participan en este proyecto. Actualmente, ya son 33 las universidades Iberoamericanas que ofrecen formación a través de MiriadaX (Universia, 2014). Los procedentes de América son la Universidad Abierta para Adultos, Universidad Blas Plascal, Universidad de Ibagué, Universidad de Puerto Rico, Universidad Tecnológica de Pereira, National University College, Universidad Católica Santo Toribio de Mogrovejo y Universidad de San Martín de Porres.

Desde su fecha de lanzamiento, las universidades han contribuido a la publicación de 153 cursos y cuenta con una comunidad de 990 profesores (Universia, 2014).



Figura 3.18: Universidades asociadas a Miriada X.  
Fuente: (Miriada X, 2014)

- **Conócenos**

En el espacio Conócenos se ofrecen conceptos sobre la filosofía que impera en el proyecto, las publicaciones del blog, las normas de trabajo en MiriadaX y cómo unirse al proyecto.

- **Soporte**

Es el espacio en donde los estudiantes pueden acceder para consultar preguntas frecuentes o pedir ayuda al centro de ayuda de la plataforma con algún problema en el desarrollo de los cursos.

### Certificados ofrecidos por MiriadaX

MiriadaX ofrece dos tipos de certificados para los estudiantes que cumplan con los requisitos establecidos para aprobar un MOOC alojado en esta plataforma. Estos son:

- **Gratuito:** Denominado como certificado de participación, y se lo consigue automáticamente desde la plataforma una vez que se ha superado el 75% del curso.



Figura 3.19: Certificado gratuito emitido por MiriadaX.

Fuente: [https://www.miriadax.net/web/informacion\\_cientifica/blog](https://www.miriadax.net/web/informacion_cientifica/blog)

- **De pago:** Denominado certificado de superación, y se puede conseguir al alcanzar el 100% del curso, siempre y cuando se realice el pago que haya establecido MiriadaX y la universidad encargada de impartir el MOOC.



Figura 3.20: Certificado de pago emitido por MiriadaX.

Fuente: <http://infinitasdimensiones.blogspot.com/2013/05/miriadax-nuevos-certificados-premios-y.html>

La principal diferencia entre estos tipos de certificados es que el certificado de pago tendrá la imagen de la universidad que lo imparte y la firma del tutor del MOOC (Octeto, 2012).

### 3.2.4 EdX

Es una plataforma de cursos en línea masivos y abiertos (MOOC) fundada por el Instituto Tecnológico de Massachusetts y la Universidad de Harvard en mayo de 2012 para hospedar cursos online de nivel universitarios de una amplio rango de disciplinas para todo el mundo sin costos para propiciar la investigación y el aprendizaje. EdX tiene más de 2 millones de usuarios. Cada una de estas dos instituciones contribuye con 30 millones de dólares de recursos para este proyecto sin ánimo de lucro. El curso prototipo, Circuitos y Electrónica, comenzó en diciembre de 2011, a través de MITx, el programa online y masivo del MIT. Actualmente hay 47 institutos, organización sin fines de lucro, corporaciones y organizaciones internacionales que ofrecen o planean ofrecer cursos en el sitio web de edX (Colaboradores de Wikipedia, 2014).



*Figura 3.21: Logotipo de edX.*

*Fuente: <https://www.edx.org/>*

El entorno en línea ofrece una potente plataforma para llevar a cabo estos experimentos, la exploración de cómo aprenden los estudiantes y cómo pueden enseñar los profesores ha sido favorecido por el uso de una variedad de nuevas herramientas y técnicas como videos en línea, material didáctico en la red, autoevaluaciones con retroalimentación, etc.

#### Características

- Experiencia de usuario superior a la de las plataformas tradicionales, en términos de usabilidad, flexibilidad, velocidad y funcionalidad.
- Adaptabilidad a múltiples dispositivos y tipos de pantallas.
- Evolución y mejoramiento permanente de la tecnología.
- Tecnología respaldada por las mejores universidades del planeta (Edunext, 2014).

#### Componentes

El ecosistema de soluciones de open edX, disponible bajo una licencia AGPL, está desarrollado en Python, Javascript, Ruby y Node.js, e incluye:

- **Edx-Platform.** Un sistema LMS o plataforma de gestión de cursos.
- **Studio**, la herramienta para la creación de cursos y gestión de contenidos
- **Discern**, una API de calificación automatizada para respuestas escritas que funciona en combinación con el motor de evaluación con inteligencia artificial EASE (enhanced AI scoring engine).
- **Insights**, un framework para el análisis de la información recolectada durante el proceso de aprendizaje.
- **CS comments service**, una aplicación de Ruby que soporta los foros discusión con funcionalidad de votaciones y validación de comentarios.
- **XQueue** que define la interface a través de la cual el LMS puede comunicarse con servicios de calificación externos.
- **XServer** para la ejecución de secuencias de código enviadas por los estudiantes.
- **XBlock**, una API adicional para la integración de contenidos en los cursos.

#### Arquitectura de EdX

La arquitectura de la plataforma edX cuenta con varios componentes, tal y como lo podemos observar en la figura 3.22.

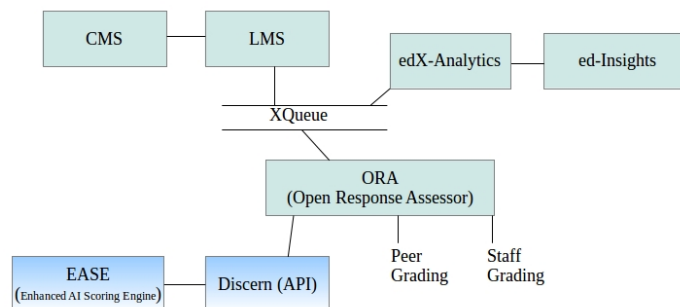


Figura 3.22: Arquitectura de la plataforma EDx.

Fuente: (Colaboradores de Diario Linux, 2014)

La plataforma está compuesta básicamente por:

- Repositorio de Cursos
- Interfaz con sistemas externos de calificación
- Bibliotecas de código y módulos para calificación
- Entorno para ejecución de programas

### Repositorio de cursos

Actualmente la plataforma del LMS (Learning Management sistema de gestión de aprendizaje) está basada en XModules (componentes de cursos), pero está siendo migrada a XBlock (arquitectura de componentes para construir cursos) (Meza Escallón, 2014).

**Configuration** suministra una forma simple pero flexible de levantar una instancia de la plataforma edX completamente configurada y lista para usarse.

### Interfaz con sistemas externos de calificación

**XQueue** define una interfaz entre el LMS y sistemas externos que provean la calificación de exámenes y trabajos enviados por los estudiantes. Esto permite que se usen los módulos de calificación que ya trae edX o que uno desarrolle o adapte su propio mecanismo personalizado de calificación y lo conecte con el LMS (Meza Escallón, 2014).

### Calificación

Como la idea de esta plataforma es que los cursos que se publiquen en esta plataforma estén abiertos para una comunidad enorme (alrededor de 100.000 participantes simultáneos) es necesario contar con mecanismos para calificar que no dependen de la intervención manual de uno o varios profesores. De esta manera, no solo se evalúan los cuestionarios de selección única o múltiple, sino también respuestas abiertas. Para eso se cuenta con módulos como:

**edx-ora** (Open Response Assessor) recibe trabajos enviados desde el LMS a través de la interfaz XQueue, los pasa ya sea por un motor de calificación basado en aprendizaje automático, un sistema de calificación manual por pares o por profesores, según corresponda y devuelve el resultado al LMS.

**Discern** es un API que permite el uso de un motor de clasificación de textos basado en aprendizaje automático. Si uno quiere desarrollar su propio motor de calificación, debe implementarlo dentro de esta interfaz para comunicarse con el LMS. Una forma de hacerlo es basándose en bibliotecas de funciones como EASE.

**EASE** (Enhanced AI Scoring Engine) no es un módulo sino una biblioteca de funciones que permite alimentar mecanismos de clasificación de contenido textual basado en aprendizaje

automático. Es útil para calificar exámenes que no se ajustan a un formulario de selección múltiple, como ensayos o preguntas de respuesta abierta. Suministra funciones que asignar una nota a texto arbitrario y predictores numéricos, buscando predecir objetivos a partir de valores arbitrarios de forma escalable y con alto desempeño.

#### Entorno para ejecución de programas

**CodeJail** gestiona la ejecución de programas en entornos seguros de prueba. Está diseñado en principio para ejecución de código Python, pero también se puede usar para otros lenguajes.

**XServer** recibe código fuente de programas enviados por los estudiantes desde el LMS y lo ejecuta usando sistemas de calificación, los cuales deben ser desarrollados aparte.

### 3.2.5 OpenCourseWare (OCW)

OpenCourseWare, conocido también como OCW, es un proyecto que nace en el año 2000 por parte del Instituto Tecnológico de Massachusetts (MIT), cuyo objetivo fue ampliar su oferta educativa para introducirse en la enseñanza virtual. El MIT fue más allá de crear un campus virtual como el que tienen la mayoría de universidades alrededor del mundo; el MIT optó por hacer públicos sus cursos, materiales docentes y propuestas académicas por medio de proyecto OpenCourseWare (Romo Uriarte, Benito Gómez, Portillo Berasaluce, & Casquero Oyarzabal, 2014).

El principal objetivo de OpenCourseWare es promover y desarrollar la educación compartiendo los recursos docentes con:

- Otros educadores que los puedan reutilizar para sus enseñanzas.
- Estudiantes, graduados y público en general que deseen mejorar su conocimiento (Universidad Politécnica de Madrid, 2014).

Desde finales del 2004, el MIT ha impulsado por medio del proyecto OCW, la formación de grupos de universidades que desarrollaran sus propios “OCW”, por lo que se creó el OpenCourseWare Consortium, cuyo principal objetivo es la difusión y coordinación de la idea de producción y distribución de contenidos educativos abiertos (Romo Uriarte, Benito Gómez, Portillo Berasaluce, & Casquero Oyarzabal, 2014).

En la actualidad, son miembros del OCW Consortium diversas instituciones educativas de los 5 continentes entre los que tenemos a la Universidad de Michigan, Universidad de Cadiz, Universidad de Valencia, MIT, Standford, Universidad del Valle-Cali, entre otras (Pernías Peco & Marco Such, 2007).



Figura 3.23: Logotipo de Open Course Ware  
Fuente: <http://www.occonsortium.org/courses/>



### Características de OCW

Los OCW son espacios web que contienen materiales docentes para la formación superior. Las principales características de este proyecto son:

- Los recursos didácticos publicados en un OCW se organizan en asignaturas o cursos, es decir, se consideran dos aspectos importantes:
  - El estudiante accede al sitio y se matricula en una asignatura o curso para tener acceso al conjunto de materiales asociados a ella.
  - El material está organizado en categorías: programa de la asignatura, lecturas obligatorias, material de clase, ejercicios, guía de aprendizaje, etc.
- Los docentes garantizan que el material publicado en el OCW es original y que los derechos legales del mismo permiten que sean publicado y reutilizados de forma gratuita, sin infringir los copyrights de otras personas.
- Son accesibles universalmente a través de Internet, sin limitaciones geográficas, ni exclusión de usuarios ni necesidad de registro alguno.
- No exigen requisitos técnicos a parte de un navegador Web (Universidad Politécnica de Madrid, 2014).

Cabe recalcar que los OCW no son sitios de educación a distancia, por lo que no se puede acceder a los docentes por medio de la plataforma. Un OCW no dispone de un correo electrónico, foros u otros medios de interacción entre el profesor y el alumno. Además, el estudiante no podrá reclamar acreditación alguna o reconocimiento por parte de la Institución que publique los contenidos (Universidad Politécnica de Madrid, 2014).

### Estructura

Los OCW están formados por asignaturas o cursos, los cuales tienen la siguiente estructura:

1. Guía de aprendizaje de la asignatura: Tiene como objetivo orientar y guiar al estudiante a través del curso. Está conformado por:
  - a. Objetivos formativos: Contiene las competencias que se pretenden alcanzar con el curso o asignatura.
  - b. Estructura: El docente se encargará de explicar cómo ha organizado el contenido de la asignatura y como deben los estudiantes utilizar el material docente.
2. Temas de la asignatura: Contiene los siguientes apartados:
  - a. Introducción
  - b. Objetivos
  - c. Esquema
  - d. Desarrollo
  - e. Resumen
3. Material Educativo: El material educativo que consta en la asignatura debe estar organizado en unidades didácticas y dentro de cada una de ellas por tipo de material (teoría, prácticas, ejemplos, videos, etc.) (Universidad Politécnica de Valencia, 2012).



### 3.3 Diagramas de clase de plataformas MOOC

Diagrama de clases de un curso en Google Course Builder

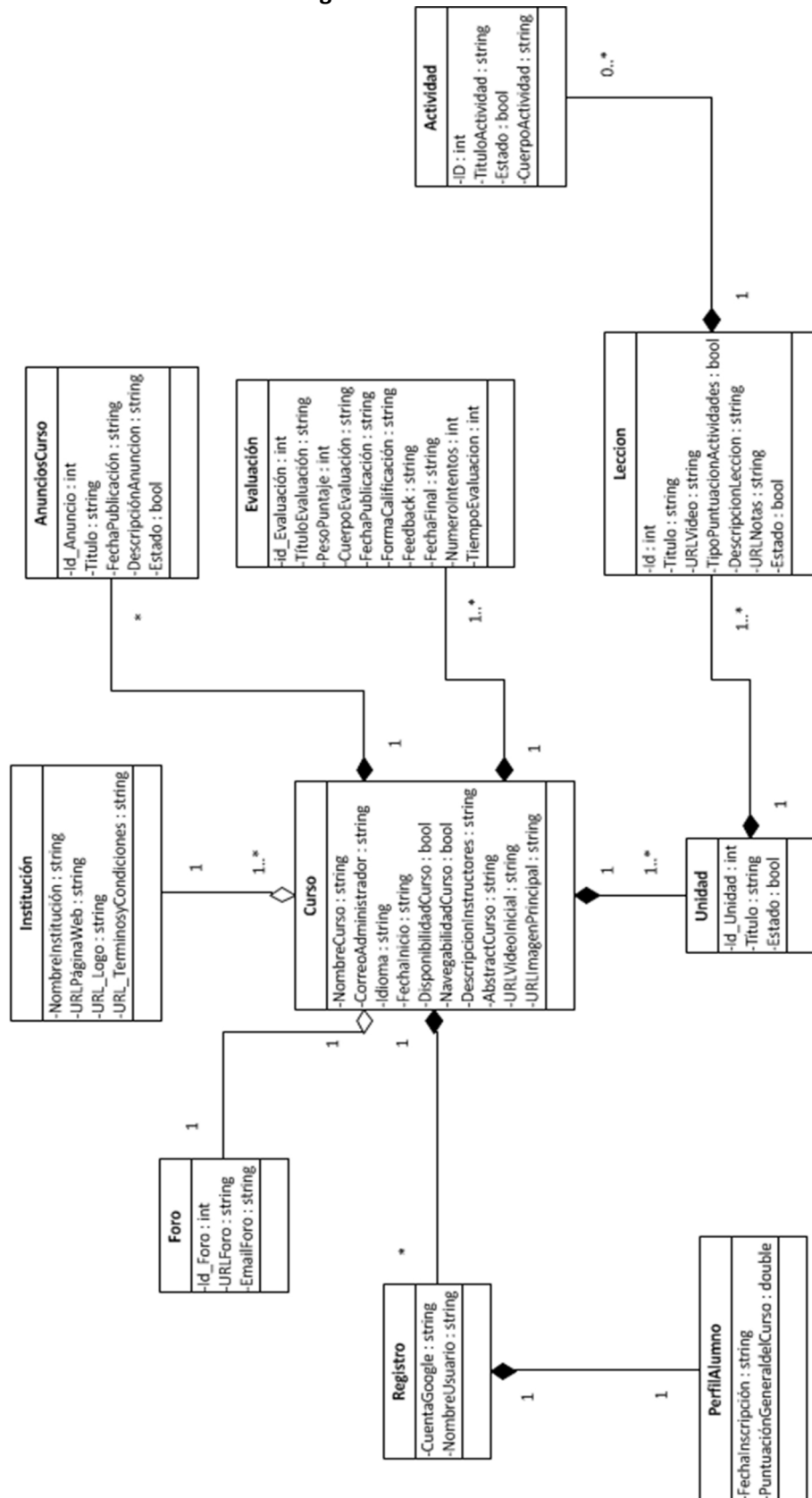


Figura 3.24: Diagrama de clases de un curso en Google Course Builder.  
Fuente: Elaboración Propia



## Diagrama de clases de un curso en Coursera

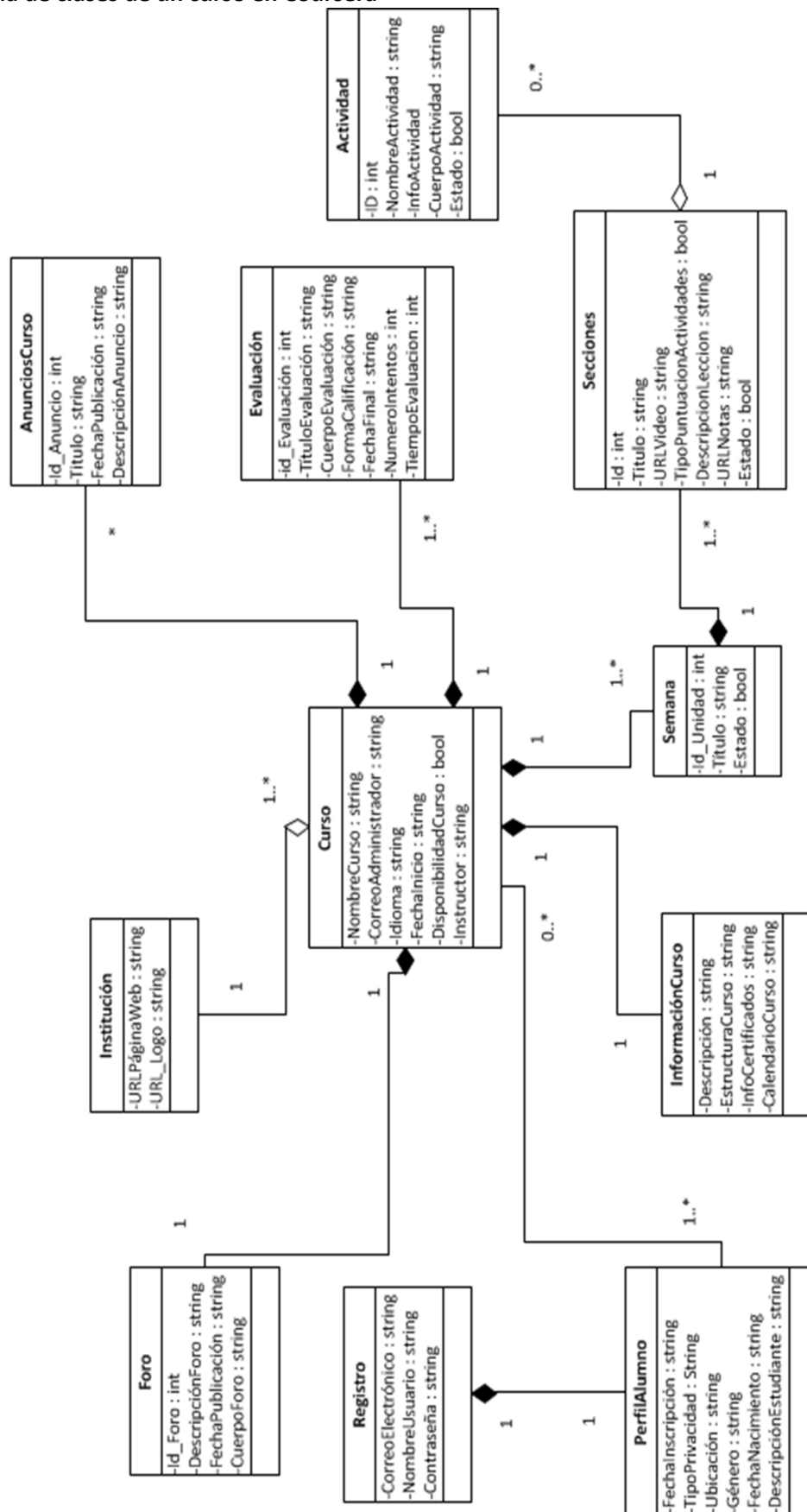


Figura 3.25: Diagrama de clases de un curso en Coursera.

Fuente: Elaboración Propia

## Diagrama de clases de un curso en MiriadaX

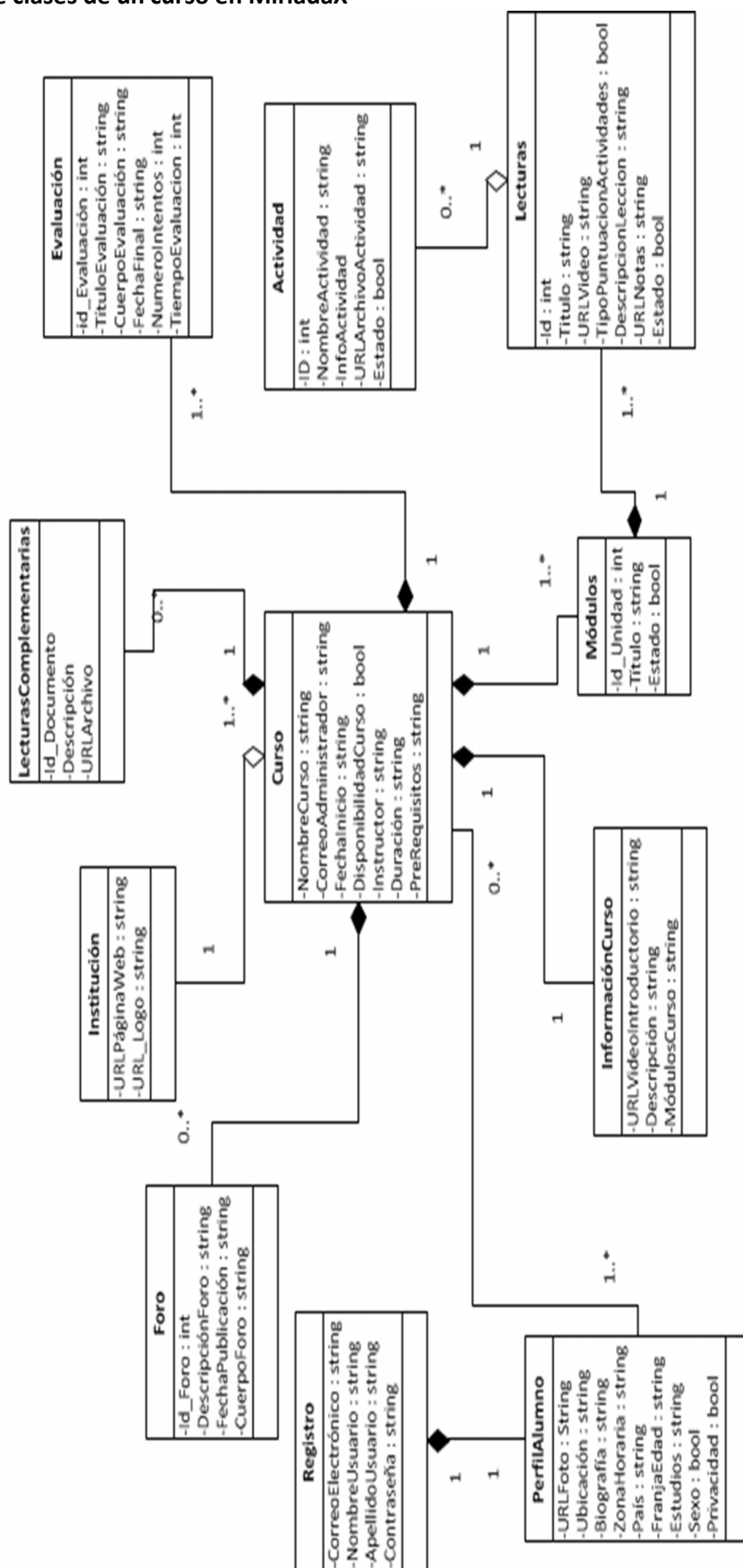


Figura 3.26: Diagrama de clases de un curso en MiriadaX  
Fuente: Elaboración Propia

## Diagrama de clases de un curso en EdX

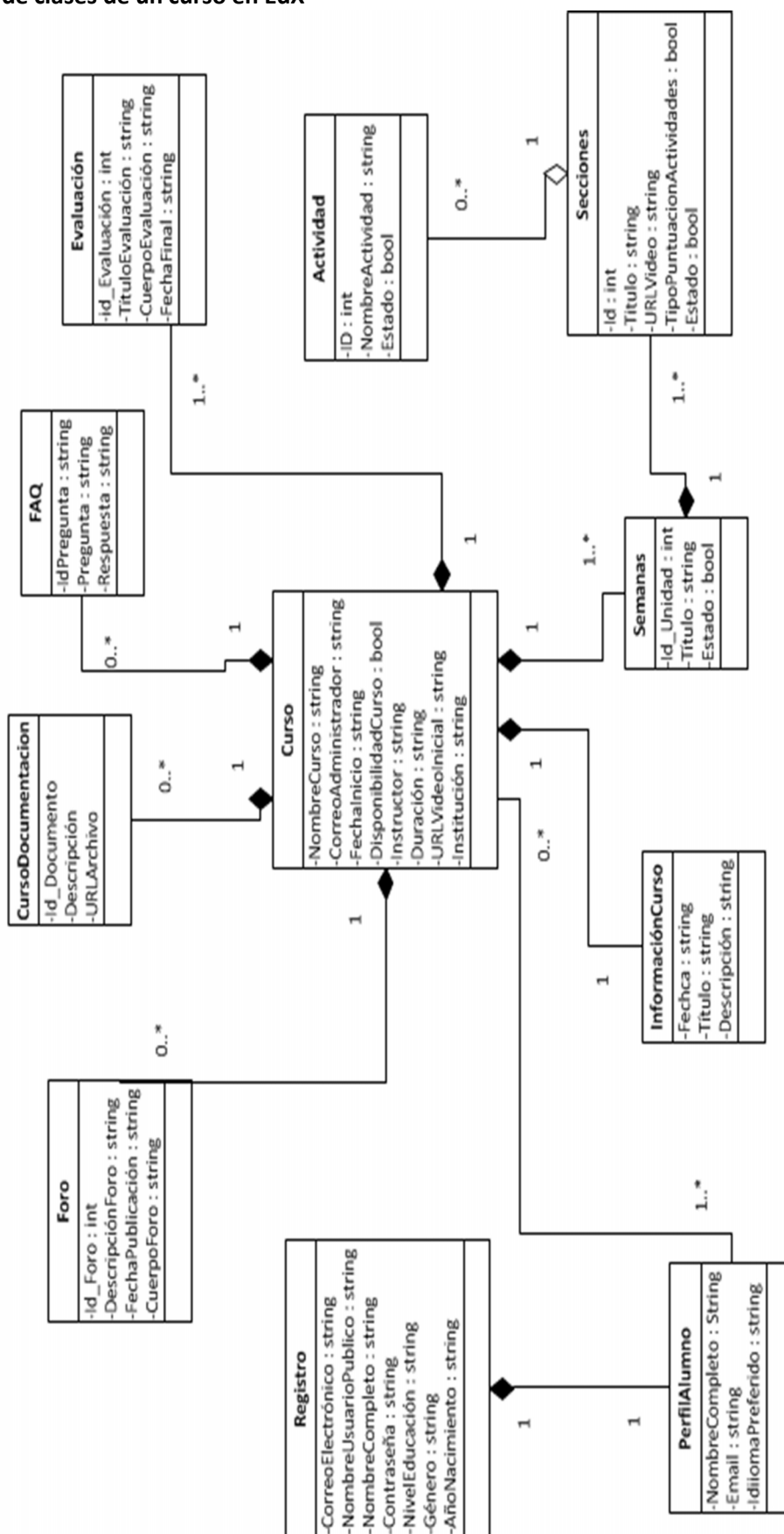


Figura 3.27: Diagrama de clases de un curso en EdX

Fuente: Elaboración Propia



Diagrama de clases de un curso en Open Course Ware

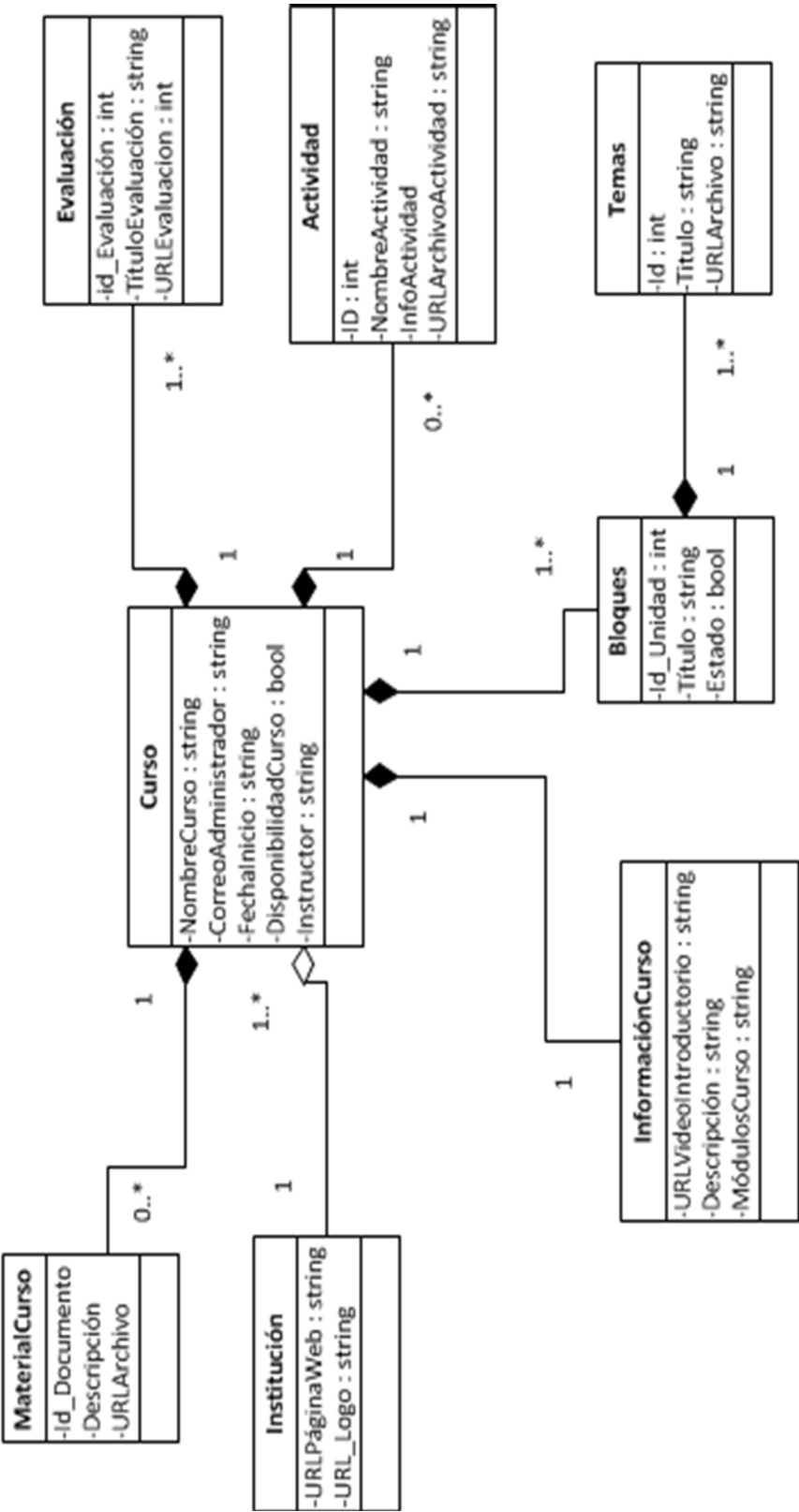


Figura 3.28: Diagrama de clases de un curso en Open Course Ware  
Fuente: Elaboración Propia

## 3.4 Cuadro comparativo entre plataformas MOOC

En base a la descripción de cada una de las plataformas MOOC que se realizó en la sección anterior, se ha realizado un cuadro comparativo de las características que cada una de ellas posee.

Para este esto, se han considerado los elementos que ofrecen cada una de las plataformas desde dos enfoques:

- **Elementos del curso:** Dentro de este aspecto se evalúan las opciones que cada una de las plataformas nos ofrecen para el diseño del MOOC, los mismos que se han obtenido de los diagramas de clases revisados anteriormente. Los elementos a evaluar son:

CARACTERÍSTICA	DESCRIPCIÓN
<b>Perfil de Usuario</b>	La plataforma, por medio del registro de usuarios, permite crear un espacio en donde los estudiantes tengan acceso a sus datos personales, fecha de inscripción en el MOOC, registro de calificaciones, etc.
<b>Foro</b>	La plataforma permite crear un espacio en donde los estudiantes puedan interactuar tanto con el tutor del MOOC como con los demás estudiantes que se encuentran matriculados en el curso.
<b>Actividades</b>	La plataforma permite que los estudiantes realicen actividades con las que se puedan verificar los conocimientos adquiridos en el curso y que permitan dar retroalimentación a los contenidos que así lo necesiten.
<b>Evaluaciones</b>	La plataforma permite al tutor evaluar los conocimientos adquiridos por los estudiantes en un momento de tiempo dado en el curso, asignándole un puntaje a cada evaluación realizada.
<b>Material Adicional</b>	La plataforma permite al tutor facilitar al estudiante material adicional como lecturas, enlaces, archivos, etc.
<b>Videos</b>	La plataforma permite el uso de videos para impartir los contenidos a los estudiantes.

Tabla 3.3: Descripción de las características a evaluar en el enfoque "Elementos del curso".

Fuente: Elaboración propia

- **Aspectos técnicos:** Dentro de este aspecto se evalúan las características de la plataforma como recursos informáticos, diseñado para apoyar los procesos educativos. Los elementos a evaluar en este enfoque son:

CARACTERÍSTICA	DESCRIPCIÓN
<b>Versión Móvil</b>	Adapta el MOOC para que pueda ser visualizado desde cualquier dispositivo móvil (teléfonos, tablets, etc.)
<b>Registro de usuarios dinámico</b>	La plataforma permite establecer las preguntas que se les realizará a los estudiantes del MOOC al momento del registro.
<b>Gratuito</b>	Para poder publicar un curso en la plataforma no se requiere pagar.

<b>Acceso a código fuente</b>	La plataforma permite modificar el código fuente del MOOC para personalizarlo acorde a nuestras necesidades.
<b>Uso de servidores externos para alojar el MOOC</b>	La plataforma permite alojar el MOOC en cualquier servidor externo propio.
<b>Definición del lenguaje del MOOC</b>	Permite cambiar de idioma a la estructura del MOOC de forma fácil, sin tener que para ello modificar todo el código fuente.

Tabla 3.4: Descripción de las características a evaluar en el enfoque "Aspectos Técnicos"

Fuente: Elaboración propia

Establecidos los criterios que serán tomados en cuenta, en la tabla 3.5 se puede observar un cuadro comparativo de las plataformas MOOC seleccionadas, tomando como base, el análisis realizado en la sección anterior.

<b>Elementos del curso</b>	<b>Google Course Builder</b>	<b>EdX</b>	<b>Coursera</b>	<b>Open Course Ware</b>	<b>MiriadaX</b>
<b>Perfil de Usuario</b>	SI	SI	SI	NO	SI
<b>Foro</b>	SI	SI	SI	NO	SI
<b>Actividades</b>	SI	SI	SI	SI	SI
<b>Evaluaciones</b>	SI	SI	SI	SI	SI
<b>Material Adicional</b>	SI	SI	SI	SI	SI
<b>Videos</b>	SI	SI	SI	NO	SI
<u>Aspectos técnicos</u>					
<b>Versión Móvil</b>	SI	SI	SI	NO	SI
<b>Registro de usuarios dinámico</b>	SI	NO	NO	NO	NO
<b>Gratuito</b>	SI	SI	NO	NO	NO
<b>Acceso a código fuente</b>	SI	SI	NO	NO	NO
<b>Uso de servidores externos para alojar el MOOC.</b>	SI	NO	NO	NO	NO
<b>Definición del lenguaje del MOOC</b>	SI	NO	NO	NO	NO

Tabla 3.5: Cuadro comparativo de las características de las plataformas MOOC.

Fuente: Elaboración propia

Como se puede observar en la tabla 3.5, las plataformas MOOC elegidas comparten algunas características como el uso de actividades de aprendizaje, evaluación de contenidos, el uso de material adicional como PDF, libros digitales, etc., y cada una tiene otras que las diferencian como el uso de foros para fomentar la participación de los estudiantes, el uso de videos para la transmisión de conocimiento, lo que nos permite determinar las fortalezas y debilidades de cada una ellas.

Una vez analizadas las características de cada una de las plataformas, se procede a la comparación de las plataformas analizadas en base a las Normas ISO 25010:2010. Se toma en cuenta las características y subcaracterísticas de las que se conforma esta norma (International Organization for Standardization, 2010).

En la tabla 3.6, se puede observar una descripción de cada una de las características y subcaracterísticas de la norma ISO mencionada.

**CARACTERÍSTICAS Y  
SUBCARACTERÍSTICAS**
**DESCRIPCIÓN**

Funcionabilidad	<b><i>Grado en el que un producto o sistema provee las funciones que cumplen con las necesidades expresadas.</i></b>
Integridad funcional	Grado en el que un conjunto de funciones cubre todas las tareas y objetivos específicos de los usuarios.
Corrección funcional	Grado en el que un producto o sistema proporciona los resultados correctos con el grado necesario de precisión.
Adecuación funcional	Grado en el que las funciones facilitan el cumplimiento de tareas y objetivos específicos.
Eficiencia	<b><i>Resultados respecto a la cantidad de recursos utilizados bajo unas condiciones establecidas</i></b>
Comportamiento del tiempo	Grado en el que los tiempos de respuesta y procesamiento se dan. Tasas de rendimiento de un producto o sistema.
Utilización de recursos	Cantidades y tipos de recursos utilizados por un producto o sistema, al momento de ejecutar funciones para cumplir con requisitos. Los recursos humanos se incluyen en esta subcaracterística.
Capacidad	Grado en que los límites máximos de un parámetro de un producto o sistema cumplen, con los requisitos. Los parámetros pueden ser el número de usuarios concurrentes, el ancho de banda de la comunicación, número de transacciones en un instante de tiempo, y tamaño de la base de datos.
Compatibilidad	<b><i>Grado en que un producto, sistema o componente, puede intercambiar información con otros productos, sistemas o componentes, y realizar las funciones requeridas mientras comparten el mismo hardware o entorno de software.</i></b>
Co-existencia	Grado en el cual un producto puede llevar a cabo sus funciones de manera eficiente aunque comparta al mismo tiempo recursos y entornos con otros productos, sin impactar de forma negativa a otro producto.
Interoperabilidad	Grado en el que dos o más sistemas, productos o componentes pueden intercambiar información y utilizar la información que se ha intercambiado.
Usabilidad	<b><i>Grado en el que un producto o sistema puede ser usado por determinados usuarios para conseguir objetivos específicos con eficiencia, eficacia y satisfacción en un contexto específico.</i></b>
Facilidad de reconocimiento	Grado en el cual los usuarios pueden reconocer si un producto o sistema es adecuado para sus necesidades. Depende de la habilidad que tenga el usuario para reconocer las funcionalidades del sistema o disponibilidad y claridad del soporte documental.
Facilidad de aprendizaje	Grado de facilidad en el que un producto o sistema puede ser utilizado por un usuario, de forma eficaz, eficiente, libre de riesgo y a satisfacción en un contexto de uso específico.
Operatividad	Grado en el que un producto o sistema tiene atributos que lo hacen fácil de manejar y controlar.

Protección de errores de usuario	Grado en que el sistema protege a los usuarios para evitar errores.
Estética de la interfaz de usuario	Grado en el que una interfaz de usuario agrada y facilita la interacción con el usuario.
Accesibilidad	Grado en que un producto o sistema puede ser utilizado por personas con la más amplia gama de características y capacidades para alcanzar un objetivo establecido en un contexto de uso específico.
Fiabilidad	<b>Grado en el que un producto, sistema o componente, realiza funciones bajo condiciones específicas para un período de tiempo dado. Puede ser afectada por cualquiera de los siguientes factores: disponibilidad, fiabilidad (tolerancia a fallas y Recuperabilidad), seguridad (confidencialidad e integridad), mantenibilidad, durabilidad y soporte de mantenimiento.</b>
<b>Madurez</b>	Grado en el que un sistema satisface las necesidades de funcionamiento normal.
Disponibilidad	Grado en el que un sistema, producto o componente está operativo y accesible cuándo se requiera para su uso.
Tolerancia a fallas	Grado en el que un sistema, producto o componente funcionan como está diseñado, a pesar de la presencia de fallas en el hardware o software.
Recuperabilidad	Grado en el que, después de presentarse una interrupción o falla inesperada (ejemplo: apagado forzoso de un equipo), un producto o sistema puede recuperar los datos afectados y restablecer su funcionamiento a estado normal.
Seguridad	<b>Grado en el que un producto o sistema protege la información y los datos, para que las personas u otros productos o sistemas puedan accederlos o no de acuerdo a los tipos y niveles de autorización establecidos.</b>
Confidencialidad	Grado en el que un producto o sistema garantiza que los datos sean accesibles para las personas autorizadas a tener acceso.
Integridad	Grado en el que un sistema, producto o componente impide el acceso no autorizado o modificación de programas o datos del computador.
No-repudio	Grado en el que las acciones o eventos se pueden demostrar, de modo que no puedan ser repudiadas o desestimadas más tarde.
Responsabilidad	Grado en el que las acciones de una entidad, se remontan únicamente a la entidad.
Autenticidad	Grado en el que la identidad de un sujeto o de los recursos, puede ser comprobada.
Mantenibilidad	<b>Grado de eficacia y eficiencia con la que un producto o sistema es modificado por las personas a cargo de su mantenimiento. Las modificaciones pueden incluir correcciones, mejoras o adaptaciones del software debido a cambios en el entorno o requerimientos y especificaciones funcionales.</b>
Modularidad	Grado en el que un sistema o programa se compone de componentes específicos, donde los cambios realizados en uno de sus componentes tiene un impacto mínimo en los demás.



Reusabilidad	Grado en el que un componente puede ser utilizado en más de un sistema, o en la construcción de otros componentes.
Analizabilidad	Grado de eficacia y eficiencia con la que es posible evaluar el impacto de un producto o sistema respecto a un cambio que se pretenda hacer en una o varias de sus partes, o para el diagnóstico acerca de las causas de fallas o deficiencias de un producto, para identificar las partes que se deban cambiar.
Modificabilidad	Grado en el que un producto o sistema puede ser eficaz y eficiente al aplicar modificaciones para la corrección de fallas, sin degradar la calidad de los productos existentes.
Facilidad de pruebas	Grado de eficacia y eficiencia con que los criterios de pruebas se puedan establecer en un producto, sistema o componente, a fin de verificar el cumplimiento de dichos criterios.
Portabilidad	<b>Grado de eficacia y eficiencia de un sistema, producto o componente transferido, sin importar el hardware, software u otro entorno operativo.</b>
Adaptabilidad	Grado en el que un producto o sistema es adaptado eficaz y eficientemente a diferentes entornos operativos, de hardware, software u otros.
Instalabilidad	Grado de eficacia y eficiencia con la que puede ser instalado un producto o sistema en un ambiente específico.
Intercambiabilidad	Grado en el que puede ser reemplazado un producto por otro para la misma finalidad en el mismo entorno.

Tabla 3.6: Características y subcaracterísticas de la Norma ISO 25010:2010.

Fuente: (International Organization for Standardization, 2010)

Luego de detallar cada una de las características a evaluar, se procede a establecer una métrica por cada definición así como su escala y criterios de interpretación adecuados que brindarán mayor claridad sobre los pesos dados al momento de realizar su calificación y revisión posterior.

Por su relevancia, la métrica definida debe ser cuantificable, dado que es necesario determinar el grado de cumplimiento de software que se esté evaluando respecto a las características de calidad de alto nivel (Botero Villalba & Mera, 2011).

A continuación se detallan las métricas que serán tomadas en cuenta para la evaluación de las plataformas y que se han considerado son importantes para hacer la selección de una plataforma que servirá para la implementación de un MOOC.

MÉTRICA	TIPO	DESCRIPCIÓN
<b>SI: 10, NO: 0</b>	Booleana	SI: La plataforma cumple con la característica a ser evaluada. NO: Los atributos de la plataforma no son suficientes para que ésta cumpla con la especificación de la característica.
<b>ALTO: 10, MEDIO: 5, BAJO: 0</b>	Cuantificable	ALTO: La plataforma cumple con la característica totalmente. MEDIO: La plataforma cumple en parte con la característica, pero no completamente.

		BAJO: Los atributos de la plataforma no son suficientes para que ésta cumpla con la especificación de la característica.
--	--	--

Tabla 3.7: Descripción de las métricas utilizadas en la evaluación de las plataformas.

Fuente: Elaboración Propia

Se han considerado estos parámetros considerando que este proyecto de tesis tiene líneas de investigación futuras que serán explicadas en el Capítulo 6 y que el estudio realizado en este documento servirá de base para las mismas.

Una vez que se ha explicado los criterios para evaluar se procede a realizar la comparación de las plataformas MOOC, la misma que podemos observar en la tabla 3.8.

CARACTERÍSTICAS Y SUBCARACTERÍSTICAS ISO/IEC 25010:2010	MÉTRICAS	PLATAFORMAS MOOC				
		Google Course Builder	EdX	Coursera	OpenCourseWare	MiriadaX
<b>Funcionabilidad</b>						
Integridad	SI: 10, NO: 0	SI	SI	SI	SI	SI
Corrección	ALTO: 10, MEDIO: 5, BAJO: 0	ALTO	ALTO	MEDIO	BAJO	ALTO
Adecuación	SI: 10, NO: 0	SI	SI	SI	NO	SI
<b>Eficiencia</b>						
Comportamiento del tiempo	SI: 10, NO: 0	SI	SI	SI	NO	NO
Utilización de Recursos	SI: 10, NO: 0	SI	SI	SI	SI	SI
Capacidad	SI: 10, NO: 0	SI	SI	SI	SI	SI
<b>Compatibilidad</b>						
Co-existencia	SI: 10, NO: 0	SI	SI	SI	SI	SI
Interoperabilidad	ALTO: 10, MEDIO: 5, BAJO: 0	ALTO	MEDIO	BAJO	ALTO	BAJO
<b>Usabilidad</b>						
Facilidad de reconocimiento	SI: 10, NO: 0	SI	SI	SI	SI	SI
Facilidad de aprendizaje	SI: 10, NO: 0	SI	SI	SI	SI	SI
Operatividad	ALTO: 10, MEDIO: 5, BAJO: 0	ALTO	ALTO	MEDIO	MEDIO	ALTO
Protección de errores de usuario	SI: 10, NO: 0	SI	SI	SI	SI	SI
Estética de la interfaz de usuario	ALTO: 10, MEDIO: 5, BAJO: 0	ALTO	ALTO	ALTO	MEDIO	ALTO
Accesibilidad	SI: 10, NO: 0	SI	SI	SI	SI	SI
<b>Fiabilidad</b>						
Madurez	ALTO: 10, MEDIO: 5, BAJO: 0	MEDIO	ALTO	ALTO	MEDIO	MEDIO
Disponibilidad	SI: 10, NO: 0	SI	SI	SI	SI	SI
Tolerancia a fallas	SI: 10, NO: 0	SI	SI	SI	SI	SI
Recuperabilidad	SI: 10, NO: 0	SI	SI	SI	SI	SI
<b>Seguridad</b>						
Confidencialidad	ALTO: 10, MEDIO: 5, BAJO: 0	ALTO	ALTO	ALTO	MEDIO	ALTO
Integridad	SI: 10, NO: 0	SI	SI	SI	SI	SI
No-repudio	ALTO: 10, MEDIO: 5, BAJO: 0	ALTO	MEDIO	ALTO	BAJO	MEDIO
Responsabilidad	SI: 10, NO: 0	SI	SI	SI	SI	SI
Autenticidad	SI: 10, NO: 0	SI	NO	SI	SI	SI
<b>Mantenibilidad</b>						
Modularidad	SI: 10, NO: 0	SI	SI	SI	SI	SI
Reusabilidad	SI: 10, NO: 0	SI	SI	SI	SI	SI
Analizabilidad	SI: 10, NO: 0	SI	SI	SI	NO	SI
Modificabilidad	SI: 10, NO: 0	SI	SI	SI	SI	SI
Facilidad de pruebas	SI: 10, NO: 0	SI	SI	SI	SI	SI
<b>Portabilidad</b>						
Adaptabilidad	ALTO: 10, MEDIO: 5, BAJO: 0	ALTO	MEDIO	BAJO	ALTO	BAJO
Instalabilidad	ALTO: 10, MEDIO: 5, BAJO: 0	ALTO	MEDIO	BAJO	ALTO	BAJO
Intercambiabilidad	SI: 10, NO: 0	SI	SI	SI	SI	SI
<b>PUNTAJE TOTAL:</b>		<b>305</b>	<b>280</b>	<b>270</b>	<b>240</b>	<b>260</b>

Tabla 3.8: Cuadro comparativo de las plataformas MOOC en base a las características y subcaracterísticas establecidas en la Norma ISO 25010:2010.

Fuente: Elaboración propia

Una vez realizada la comparación de las cinco plataformas, se ha tomado la decisión de utilizar la plataforma Google Course Builder para el desarrollo del presente proyecto puesto que es la que mayores ventajas ha presentado en la comparación entre plataformas. Además, a pesar de ser una plataforma relativamente nueva ha tenido una aceptación muy rápida a nivel mundial (ha sido utilizada en 140 países en 15 meses), lo que demuestra que en poco tiempo se ha convertido en una potencia dentro del ámbito de los MOOCs.

Google Course Builder es una plataforma que ofrece un servidor gratuito (AppEngine) para publicar los MOOCs en caso de no poseer uno, hace uso de todas las aplicaciones que ofrece Google (YouTube, Google Analytics, Drive, etc.) por lo que el contenido del MOOC no se encuentra únicamente en un servidor, sino alojados en varios servidores y cuando se desee exportar el contenido a un servidor propio, Google mismo facilita la interfaz de usuario para poder hacerlo, esto es muy ventajoso puesto que Google Course Builder es la única plataforma que gracias a su característica de interoperabilidad permite que los MOOCs estén alojados en servidores propios y no sólo en los ofrecidos en este caso por Google.

Course Builder ofrece una plataforma que puede traducir la estructura principal del curso a más de 50 idiomas con sólo cambiar la configuración del archivo principal del MOOC. Por otra parte, permite crear registro de usuarios dinámicos, que implica que a cada curso se le puede asignar una serie de preguntas distintas que permitirá obtener información personalizada de los estudiantes para luego obtener estadísticas de los resultados del MOOC de acuerdo a nuestra línea de investigación.

Finalmente, Google Course Builder es una plataforma que nos permitirá a futuro desarrollar una plataforma propia para la Universidad de Cuenca, tomando como ejemplo a UniMOOC, una plataforma basada en Google Course Builder que ofrece MOOCs para emprendedores.

### 3.5 Recapitulación

Se han analizado las arquitecturas de diferentes plataformas y se han encontrado aspectos tanto positivos como negativos en cada una de ellas. Se ha delimitado el trabajo a cinco plataformas: Google Course Builder, EdX, Coursera, Open Course Ware y MiradaX.

Para cada una de las plataformas antes mencionadas se ha realizado un análisis que permita determinar cuál de ellas es la más apropiada para la implementación del MOOC. Este análisis estuvo basado en la descripción de características de cada una de las plataformas y en la elaboración de diagramas de clases con el fin de obtener una idea clara de la arquitectura utilizada en cada una de ellas. Finalmente, en base a un cuadro comparativo se evalúan las características de las plataformas en base a la Norma ISO 25010:2010 para establecer cuál es la más apropiada para el desarrollo de esta tesis.

Luego de realizar este proceso, se ha concluido que la plataforma idónea para el desarrollo del MOOC en la presente tesis es la de Google Course Builder, debido a sus múltiples características positivas y a los resultados obtenidos luego de la comparación con las otras plataformas.

# CAPÍTULO 4:

## Diseño e Implementación de un MOOC

### 4.1 Introducción

Una vez establecido en el capítulo anterior la plataforma que se usará para implementar el MOOC, en el presente capítulo se procederá a realizar el diseño del MOOC y a su implementación. Para esto, se ha escogido la metodología establecida en la “Guía metodológica para la planificación, diseño e impartición de MOOCs” desarrollada por la Unidad de Tecnología Educativa e Innovación Docente de la Universidad Carlos III de Madrid. Esta guía surgió en Marzo del 2014 como una respuesta a las necesidades de apoyo al profesorado de la Universidad Carlos III de Madrid que deseaban impartir un MOOC, pero no sabían cómo hacerlo de forma correcta. La universidad en mención cuenta con una gran experiencia en el área de los MOOCs, ya que los han venido ofertando al público desde el año 2010. Son estas razones las que han permitido escoger esta metodología como la más adecuada para el desarrollo del presente trabajo.

### 4.2 Metodología para la producción de un MOOC

A continuación se detallan los pasos de la metodología antes mencionada para el diseño del MOOC.

#### 4.2.1 Materia a impartirse y personas a las que está orientado

El MOOC está diseñado como base complementaria para la materia de Algoritmos, datos y estructuras, impartida en la Facultad de Ingeniería a los estudiantes de primer año de la Universidad de Cuenca; sin embargo, este curso podría ser seguido por cualquier persona que quisiera adquirir conocimientos básicos de la materia en mención.

## 4.2.2 Objetivos del curso

Los objetivos que el estudiante puede alcanzar mediante el curso son los siguientes:

- Identificar conceptos generales acerca de la programación.
- Reconocer terminologías propias de la programación.
- Aplicar los conocimientos adquiridos mediante la resolución de diversas actividades propuestas.
- Analizar problemas que necesiten de la programación para su resolución.
- Evaluar los conocimientos adquiridos a lo largo del curso.

## 4.2.3 Equipo docente

El equipo docente estará formado por dos grupos de personas:

- *Curator*: Responsable de la interacción con los estudiantes en lo que se refiere a contenidos en la plataforma.
- *Facilitator*: Encargado de interactuar con los participantes en el curso en lo relativo a su dimensión técnica (Universidad Carlos III de Madrid, 2014).

En la tabla 4.1, podemos observar los roles que cada uno de ellos cumple en el curso.

CURATORS	FACILITATORS
<b>Actúan como expertos en la materia del curso</b>	Comprobar que el curso no presenta problemas en la plataforma antes de su inicio.
<b>Son portavoces del equipo docente en la plataforma</b>	Dinamizar el foro.
<b>Resuelven dudas sobre el contenido expresadas a través del foro.</b>	Resolver dudas sobre el funcionamiento de la plataforma y velar por la reputación digital del curso.
<b>Resuelven dudas sobre las actividades y otras peticiones serias.</b>	Informar a los curators sobre cualquier problema relacionado con el contenido del curso expresado en el foro.

Tabla 4.1: Roles del equipo docente.

Tabla del autor

El rol de *curator*, en este MOOC será llevado a cabo por el Ing. Jorge Maldonado, docente de la Facultad de Ingeniería de la Universidad; mientras que el rol de *facilitators*, será llevado a cabo por los autores de esta tesis (Andrea Peralta – Christian Piedra).

## 4.2.4 Estructura del curso

El curso está conformado por tres unidades, las mismas que cuentan con varias lecciones que permitirán al estudiante llegar a los objetivos planteados al inicio del curso.

Las lecciones están estructuradas de forma que contienen los siguientes elementos:

- Una introducción teórica
- Un video explicativo
- Enlaces a lecturas complementarias.
- Una versión en texto del contenido de la lección.
- Actividades de repaso al final de la misma.

Las unidades junto con sus distintas lecciones y las evaluaciones que conformarían el MOOC se lo puede observar en la tabla 4.2.

<b>1. Introducción. Conceptos Básicos</b> 1.1. ¿Qué es programar? 1.2. ¿Qué es un Lenguaje de Programación? 1.3. Modelización de problemas del mundo real 1.4. Del problema real a su solución por computadora 1.5. Conceptos sobre software
<b>EVALUACIÓN UNIDAD 1</b>
<b>2. Algoritmos</b> 2.1. Concepto de algoritmo 2.2. Concepto de variable 2.3. Tipos de datos que almacenan las variables 2.4. Representación de un algoritmo con Diagramas de Flujo 2.5. Representación de un algoritmo con Pseudocódigo 2.6. Importancia de documentar un algoritmo 2.7. Prueba de escritorio
<b>EVALUACIÓN UNIDAD 2</b>
<b>3. Condicionales y Estructuras de Control</b> 3.1. Secuencia 3.2. Decisión 3.3. Repetición 3.4. Selección 3.5. Iteración
<b>EVALUACIÓN UNIDAD 3</b>
<b>EVALUACIÓN GLOBAL DEL CURSO</b>

Tabla 4.2: Estructura del MOOC.

Tabla del autor

A continuación, en las tablas 4.3, tabla 4.4 y tabla 4.5, se presenta la estructura de cada uno de los módulos del MOOC en las que se detalla su contenido y material a utilizar.

CONTENIDOS					
	Videos	Textos	Actividades	Evaluación	Foro
Unidad 1: Introducción. Conceptos Básicos			Al final de la lección	Al final del módulo	Durante el módulo
<b>Lección 1.1:</b> <b>¿Qué es programar?</b>	¿Qué es programar?	Versión en texto del contenido Lectura 1 Conceptos Básicos Lectura 2 Introducción a la programación	Después del video		
<b>Lección 1.2:</b> <b>¿Qué es un lenguaje de programación?</b>	Introducción a Python	Versión en texto del contenido Lectura 2 Introducción a la programación	Después del video		
<b>Lección 1.3:</b> <b>Modelización de programas</b>	Modelización de problemas	Versión en texto del contenido Lectura 1 Conceptos Básicos	Después del video		

del mundo real	del mundo real			
<b>Lección 1.4: Del problema real a su solución por computadora</b>	Del problema real a su solución por computadora	Versión en texto del contenido Lectura 1 Conceptos Básicos	Después del video	
<b>Lección 1.5: Conceptos sobre software</b>	Conceptos sobre software	Versión en texto del contenido Lectura 1 Conceptos Básicos	Después del video	

Tabla 4.3: Estructura de la unidad 1.

Fuente: Elaboración propia.

	CONTENIDOS				
	Videos	Textos	Actividades	Evaluación	Foro
Unidad 2: Algoritmos			Al final de cada lección	Al final del módulo	Durante el módulo
<b>Lección 2.1: Concepto de algoritmo</b>	¿Qué es un algoritmo?	Versión en texto del contenido	Después del video		
<b>Lección 2.2: Concepto de variable</b>	02 Curso de Programación en Python : Variables, E/S, Operaciones Básicas	Versión en texto del contenido Lectura 2 Tipos de datos simples Lectura 4 Variables, sentencias, expresiones Lectura 5 Tipos de datos	Después del video		
<b>Lección 2.3: Tipos de datos que almacena una variable</b>	Tipos de datos	Versión en texto del contenido Lectura 2 Tipos de datos simples Lectura 5 Tipos de datos	Después del video		
<b>Lección 2.4: Representación de un algoritmo con Diagramas de Flujo</b>	Diagramas de Flujo	Versión en texto del contenido Aprenda a crear Diagramas de flujo	Después del video		
<b>Lección 2.5: Representación de un algoritmo con Pseudocódigo</b>	Pseudocódigo	Versión en texto del contenido Estándar Pseudocódigo			



<b>Lección 2.6: Importancia de documentar un algoritmo</b>	Importancia de documentar un algoritmo	Versión en texto del contenido Lectura 1 Corrección de algoritmos	Después del video	
<b>Lección 2.7: Prueba de escritorio</b>		Versión en texto del contenido	Después del video	

Tabla 4.4: Estructura de la unidad 2.

Fuente: Elaboración propia

CONTENIDOS					
	Videos	Textos	Actividades	Evaluación	Foro
<b>Unidad 3: Condicionales y Estructuras de Control</b>			Al final de cada lección	Al final del módulo	Durante el módulo
<b>Lección 3.1: Secuencia</b>	Estructura Secuencial	Versión en texto del contenido Lectura 1	Después del video		
<b>Lección 3.2: Decisión</b>	Tutorial python ESTRUCTURAS CONDICIONALES	Versión en texto del contenido Lectura 1 Estándar Pseudocódigo	Después del video		
<b>Lección 3.3: Repetición</b>	Bucle Para	Versión en texto del contenido Lectura 1 Estándar Pseudocódigo	Después del video		
<b>Lección 3.4: Selección</b>	Lógica de programación - Selección múltiple	Versión en texto del contenido Lectura 1	Después del video		
<b>Lección 3.5: Iteración</b>	Bucles	Versión en texto del contenido Lectura 1	Después del video		

Tabla 4.5: Estructura de la unidad 3.

Fuente: Elaboración propia

Para revisar los contenidos, actividades y evaluaciones del curso dirigirse al Anexo 1 de este documento.

Una vez que se ha establecido la estructura interna del curso, a continuación se va a explicar la estructura externa, que será la que pueden observar los estudiantes al ingresar al MOOC.

Al ingresar a la plataforma, lo primero que verá será la página de inicio, la misma que se puede observar en la figura 4.1.



Figura 4.1: Página de Inicio del MOOC.

Fuente: Elaboración propia

Esta ventana está dividida en cuatro secciones distintas las mismas que se explican a continuación:

1. **CABECERA:** Esta sección contiene el título del curso, el logo de la Universidad y un enlace para que los estudiantes que se encuentren registrados accedan al curso.
2. **MENÚ PRINCIPAL:** Contiene los enlaces que facilitarán la navegabilidad en el curso.
  - a. **Avisos:** Contiene comunicados con los que se notifica algún hecho del curso. Al iniciar el curso, en esta opción del menú, van a encontrar un aviso de bienvenida, en el que se le notificará al estudiante cuales son los temas a tratar y el calendario sugerido para realizar el MOOC. En la figura 4.2 se puede observar un ejemplo de un aviso realizado en el MOOC.
  - b. **Curso:** Está opción permite regresar a la página principal del MOOC en cualquier momento.
  - c. **Foro:** Existirán dos foros obligatorios que el estudiante deberá contestar con el fin de que comparta los conocimientos adquiridos con los demás estudiantes del MOOC. El primer foro comprenderá los contenidos revisados en el capítulo 1 y 2, mientras que el segundo será sobre el capítulo 3. Además los estudiantes podrán exponer en este lugar las preguntas que surjan a lo largo del curso para que las mismas puedan ser contestadas por sus compañeros del MOOC o por el facilitador. En la figura 4.3 se observa la imagen del foro utilizado en el MOOC.

## Bienvenidos al Curso

2014-09-15

Estimados estudiantes, bienvenidos al curso ALGORITMOS, DATOS Y ESTRUCTURAS.

Aquí se abordarán los tres primeros capítulos del sílabo de la asignatura:

- Capítulo 1: Introducción - Conceptos Básicos
- Capítulo 2: Algoritmos
- Capítulo 3: Condicionales y Estructuras de Control.

Le recordamos que tendrá una duración de 10 horas distribuidos en dos semanas, y estará disponible del 15 de septiembre al 26 de septiembre del 2014.

Se ha realizado un calendario de los tiempos que se deben dedicar a cada unidad y evaluaciones para cumplir con éxito el curso.

ACTIVIDAD	DURACIÓN
Unidad 1: Introducción. Conceptos Básicos	2 horas
Evaluación Unidad 1	10 minutos
Unidad 2: Algoritmos	3 horas
Evaluación Unidad 2	15 minutos
Unidad 3: Condicionales y Estructura de Control	4 horas
Evaluación Unidad 3	15 minutos
Evaluación Final	20 minutos
<b>TIEMPO TOTAL</b>	<b>10 HORAS</b>

Estamos seguros que luego de este tiempo, habrá obtenido conocimientos muy importantes para su futuro.

ÉXITOS!

Figura 4.2: Aviso de Bienvenida al MOOC.  
Fuente: Elaboración propia

[PUBLICAR UNA PREGUNTA](#)
[C](#)
[Marcar todo como leído](#)
[Filtros](#)
[Ayuda](#)

**FUNDAMENTOS DE PROGRAMACIÓN**
Grupo compartido públicamente
[Unirse al grupo](#)

3 de 3 temas (1 no leído)

**FORO 2: CAPÍTULO 3 (1)**  
 Por Programación Ucuencia - 1 publicación - 1 página
 08:23

¿Cómo la abstracción facilita que los algoritmos cumplan con su objetivo?  
 Por Verónica Guamán - 3 publicaciones - 104 vistas
 21 sep

**FORO 1: CAPÍTULO 1 Y 2**  
 Por Programación Ucuencia - 78 publicaciones - 374 vistas
 21 sep

Figura 4.3: Foro del MOOC.  
Fuente: Elaboración propia.

- d. **Registro/Progreso:** El último elemento del menú corresponde a dos opciones Registro/Progreso, de acuerdo al estado del estudiante se mostrará una de las dos.
- Registro:** Cuando el estudiante ingresa por primera vez al curso, este menú le permitirá registrarse en el MOOC para tener acceso a todos los contenidos del mismo. Para el registro en el MOOC, se necesita de una cuenta de Gmail o de una institución que use el servidor de correo electrónico de Google, como es el caso de la Universidad de Cuenca. Se les solicitará que llenen una encuesta inicial, con el fin de obtener la información para el caso de estudio que se desarrollará en el siguiente capítulo. En la figura 4.4 se puede observar el cuestionario utilizado en el MOOC para el registro en el MOOC.



1. Nombres:  
Ejemplo

2. Apellidos:  
Ejemplo

3. Edad:  
24

4. Sexo:  
☐ Femenino ☒ Masculino

5. Ciudad de origen:  
Ejemplo

6. ¿Ha tomado antes un curso o materia de programación?  
☐ Si ☒ No  
En caso de que su respuesta sea afirmativa, ¿Dónde?  
☒ Colegio  
☐ Institutos  
Otros (Especifique):

7. ¿Programa en algún lenguaje de programación?  
☒ Si ☐ No  
En caso de que su respuesta sea afirmativa, ¿Cuál? Java

8. ¿Es repetidor de la asignatura?  
☐ Si ☒ No

Vaya al siguiente [enlace: http://www.estilosdeaprendizaje.es/chaeta/chaeta.htm](http://www.estilosdeaprendizaje.es/chaeta/chaeta.htm)

9. Conteste el cuestionario y anote los resultados para cada uno de los siguientes valores.  
Activo: 12 Reflexivo: 15 Pragmático: 16 Teórico: 12

[Inscribirse](#)

Figura 4.4: Cuestionario de registro de estudiantes en el MOOC.

Fuente: Elaboración propia.

- ii. **Progreso:** Permite acceder a la información del estudiante: Fecha de registro, nombre, puntuación de cada una de las evaluaciones y un promedio general del curso. Cabe recalcar que se muestran únicamente el puntaje obtenido en las evaluaciones y en el curso en general, más no el porcentaje de avance en el curso. En la figura 4.5 se puede ver la ventana del progreso de un estudiante registrado en el MOOC.

Progreso del estudiante	
Fecha de inscripción	2014-09-13
Correo electrónico	ejemplo.mooc@gmail.com
Nombre	Ejemplo
Puntuaciones de la evaluación	
Evaluación Unidad 1	0
Evaluación Unidad 2	0
Evaluación Unidad 3	0
Evaluación Final	0
Puntuación general del curso	0

Figura 4.5: Progreso del estudiante en el MOOC.

Fuente: Elaboración propia.

- DESCRIPCIÓN DEL CURSO:** En esta sección se encuentra una descripción de cuál es el objetivo del MOOC.
- CONTENIDOS:** Está sección muestra los contenidos del MOOC, el mismo que está dividido en 3 unidades, las mismas que contienen una evaluación final por unidad y 1 evaluación final que abarca los contenidos de todo el curso. En la figura 4.6 se puede observar los contenidos del MOOC en la pantalla principal.

## Contenidos

Unidad 1 - Introducción - Conceptos Básicos
○ Evaluación Unidad 1
Unidad 2 - Algoritmos
○ Evaluación Unidad 2
Unidad 3 - Condicionales y Estructuras de Control
○ Evaluación Unidad 3
○ Evaluación Final

Figura 4.6: Contenidos del MOOC.

Fuente: Elaboración propia.

- **Unidades:** En la figura 4.7 se puede observar la estructura de cada una de las unidades presenta la siguiente estructura.

Curso > Unidad 1 > Lección 2

## Unidad 1 - Introducción - Conceptos Básicos

1.1 ¿Qué es programar?

Actividad

1.2 ¿Qué es un lenguaje de programación?

Actividad

1.3 Modelización de programas del mundo real

Actividad

1.4 Del problema real a su solución por computadora

Actividad


1.5 Conceptos sobre software

Actividad

### ¿Qué es un lenguaje de programación?

Versión en texto

Un lenguaje de programación es un lenguaje formal que han sido diseñados para expresar tareas de cómputo, es decir, para escribir programas.



PARA LEER:

Introducción a la Programación, Pág. 4-5


Introducción a Python.

---

### Introducción a Python

¿Qué es Python?

- Es un lenguaje de programación interpretado.
- Fue creado por Guido Van Rossum.
- Sintaxis limpia y favorece a un código legible
- Lenguaje de programación multiparadigma ya que permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional.



0:00 / 2:14

Página anterior

Página siguiente

Figura 4.7: Estructura de las Unidades del MOOC.

Fuente: Elaboración propia.

Lo primero que encontramos al ingresar en cualquiera de las unidades es una barra de navegación que nos permite saber en qué parte del curso estamos, seguido del título de la Unidad. Esto se lo puede observar en la figura 4.8



Figura 4.8: Estructura de las unidades.

Fuente: Elaboración propia.

En la parte izquierda de la figura 4.9, se puede observar un menú con todas las lecciones y actividades que contiene esa unidad. La lección o actividad seleccionada se mostrará en color negro.

1.1 ¿Qué es programar?
Actividad
1.2 ¿Qué es un lenguaje de programación?
Actividad
1.3 Modelización de programas del mundo real
Actividad
1.4 Del problema real a su solución por computadora
Actividad
1.5 Conceptos sobre software
Actividad

Figura 4.9: Menú de lecciones y actividades de la unidad.

Fuente: Elaboración propia.

- **Lecciones:** Cada una de las lecciones de las unidades está estructurada de la siguiente manera:
  - Título de la lección.
  - Descripción breve de los contenidos.

- Sección “Para Leer”: Contiene lecturas obligatorias que el estudiante debe realizar ya que serán consideradas al momento de la evaluación.
- Video: Un video corto que explicará todo el contenido de la lección.
- Versión en texto: Permitirá al estudiante obtener el contenido de la lección en versión PDF.

Esto se lo puede observar en la figura 4.10



Figura 4.10: Estructura de las lecciones del MOOC.

Fuente: Elaboración propia.

- **Actividades:** En lo referente a las actividades, estas son un conjunto de preguntas en forma de cuestionarios que permiten a los estudiantes verificar la comprensión de los contenidos. Estas actividades no inciden en el puntaje para aprobar el MOOC, pero sirven de complemento para dar retroalimentación al aprendizaje obtenido por parte del alumno.

Cada una de las actividades consta de la siguiente estructura:

- Pregunta
- Opciones de respuesta
- Botón para verificar respuestas
- Cuadro de retroalimentación: En caso de que la respuesta dada sea correcta mostrará un mensaje afirmativo, caso contrario mostrará un mensaje de respuesta incorrecta seguido por el concepto que le ayudará a reforzar los contenidos

En la figura 4.11 se puede observar la estructura de las actividades en el MOOC.





Figura 4.11: Estructura de las actividades del MOOC.

Fuente: Elaboración propia.

- **Evaluaciones:** Las evaluaciones del curso permiten validar los conocimientos adquiridos en el MOOC. Éstas pueden ser realizadas por los estudiantes las veces que deseen, pero la calificación que queda registrada para el promedio del MOOC es la obtenida en el primer intento. El ejemplo de una de las evaluaciones utilizadas en el MOOC se la puede observar en la figura 4.12.

#### 4.2.5 Participación de los estudiantes

El MOOC, con el fin de fomentar la participación de los estudiantes ante cualquier duda que surja en el transcurso del MOOC, utilizará un foro, el mismo que estará disponible durante todo el tiempo que este habilitado el curso.

El principal objetivo del foro, es facilitar a los estudiantes una herramienta que les permita comunicarse con el curator del curso. Además, al tener acceso todos los estudiantes matriculados en el MOOC, pueden interactuar entre ellos, dando puntos de vista sobre algún tema en específico y obtener retroalimentación por parte del curator.

#### 4.2.6 Evaluación

El presente MOOC utilizará la evaluación por medio de cuestionarios, los mismos que proporcionan un sistema automatizado, que son muy útiles al momento de evaluar a cientos de participantes.

Con los cuestionarios se pretende comprobar el nivel de comprensión de los contenidos de cada una de las unidades del MOOC.

Este curso está diseñado para que se apruebe con un total de 60 puntos de los 100 disponibles. Para ello, cada una de las unidades cuenta con una evaluación de 10 preguntas y al final del curso se realizará una evaluación final que tendrá 20 preguntas que abarcarán el contenido de las tres unidades.

El puntaje de cada una de las unidades se lo puede observar en la tabla 4.6.

EVALUACIÓN	PUNTAJE
Unidad 1	20
Unidad 2	20
Unidad 3	20
Final	40
TOTAL	100

Tabla 4.6: Puntaje de evaluación del MOOC.

Tabla del autor.

EVALUACIÓN DE LA UNIDAD 1.

1. La informática es la ciencia que estudia el análisis de problemas utilizando computadoras.
  - ☐ Verdadero
  - ☐ Falso
  
2. Un programa permite resolver un problema mediante instrucciones aleatorias ejecutadas en el ordenador.
  - ☐ Verdadero
  - ☐ Falso
  
3. Un algoritmo es:
  - ☐ Secuencia de pasos e instrucciones que producen un resultado deseado y es finito
  - ☐ Secuencia de pasos e instrucciones con los cuales no siempre se alcanza el resultado deseado
  - ☐ Secuencia de pasos e instrucciones infinitas.
  - ☐ Secuencia de pasos e instrucciones que no siempre alcanzan el resultado deseado
  
4. Las etapas en la resolución de un problema son:
  - ☐ Análisis, Especificación, Escritura, Verificación
  - ☐ Análisis, Diseño, Especificación, Escritura
  - ☐ Análisis, Implementación, Escritura, Verificación
  - ☐ Análisis, Diseño, Especificación, Escritura, Verificación
  
6. Un programa es un conjunto de instrucciones. Cuando se escribe por ejemplo las directrices para llegar a la casa de un amigo, se está escribiendo un programa.
  - ☐ Verdadero
  - ☐ Falso
  
7. Una especificación es el proceso de analizar los problemas del mundo real y determinar de forma clara y concreta un objetivo.
  - ☐ Verdadero
  - ☐ Falso
  
8. Python:
  - ☐ Es un lenguaje de programación interpretado, con tipado dinámico sólo para plataformas Windows.
  - ☐ Es un lenguaje de programación con tipado estático multiplataforma.
  - ☐ Es un lenguaje de programación interpretado, con tipado dinámico y multiplataforma
  
9. Especificar un problema significa establecer en forma unívoca su contexto, precondiciones y el resultado esperado.
  - ☐ Verdadero
  - ☐ Falso
  
10. A partir de un problema se debe realizar un proceso de abstracción.
  - ☐ Verdadero
  - ☐ Falso

Enviar respuestas

Figura 4.12: Estructura de las evaluaciones.  
Fuente: Elaboración propia.

## 4.2.7 Herramientas de comunicación

Como medio principal de comunicación entre alumnos y el equipo docente que se encuentra detrás del MOOC se habilitará un foro, el mismo que permitirá a los estudiantes presentar sus inquietudes al mismo tiempo que sean ellos mismo quienes puedan dar soluciones a estas inquietudes o en caso de que ninguno de ellos pueda dar una solución sea el equipo docente el que se encargue de dar una solución.

## 4.2.8 Calendario

El presente MOOC está diseñado para que el estudiante lo lleve a cabo de forma autónoma, es decir, sin necesidad de que exista un profesor que guíe el aprendizaje.

El tiempo de duración del MOOC es de 2 semanas, con una dedicación de 5 horas semanales (10 horas en total).

El tiempo que debe dedicar a cada una de las unidades y evaluaciones lo detallamos en la siguiente tabla:

ACTIVIDAD	DURACIÓN
<b>Unidad 1: Introducción. Conceptos Básicos</b>	2 horas
<b>Evaluación Unidad 1</b>	10 minutos
<b>Unidad 2: Algoritmos</b>	3 horas
<b>Evaluación Unidad 2</b>	15 minutos
<b>Unidad 3: Condicionales y Estructura de Control</b>	4 horas
<b>Evaluación Unidad 3</b>	15 minutos
<b>Evaluación Final</b>	20 minutos
<b>TIEMPO TOTAL</b>	<b>10 HORAS</b>

*Tabla 4.7: Tiempo de dedicación al MOOC.*

*Tabla del autor.*

## 4.3 Implementación del MOOC

En la siguiente sección se detallan todos los pasos que se siguieron para la implementación del MOOC.

### 4.3.1 Documentación técnica

A continuación se detallan las tecnologías utilizadas para la implementación del MOOC.

#### 4.3.1.1 HTML

HTML es un lenguaje que hace posible presentar información en Internet, más concretamente, HTML es el lenguaje en el que se “escriben” las páginas web.

El lenguaje HTML es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin fines de lucro llamado World Wide Web Consortium<sup>17</sup>, más conocido como W3C. Al tratarse de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma similar en cualquier navegador de cualquier sistema operativo.

<sup>17</sup> El World Wide Web Consortium (W3C) es una comunidad internacional donde las organizaciones miembro conjuntamente con el público en general trabajan conjuntamente para desarrollar estándares Web. Liderado por el inventor de la Web, Tim Berners-Lee y el director ejecutivo (CEO) Jeffrey Jaffe, la misión del W3C es guiar la Web hacia su máximo potencial.

Desde su creación, el lenguaje HTML ha pasado de ser un lenguaje utilizado exclusivamente para crear documentos electrónicos a ser un lenguaje que se utiliza en muchas aplicaciones electrónicas como buscadores, tiendas online y banca electrónica (Eguiluz , 2013).

#### 4.3.1.2 CSS

Las hojas de estilo en cascada o CSS por sus siglas en inglés (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS se utiliza para dar estilo a documentos HTML, separando el contenido de la presentación. Los *estilos* definen la forma de mostrar los HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en el CSS afectará a todas las páginas vinculadas a ese CSS en las que aparezca ese elemento.

CSS funciona en base a reglas, es decir, declaraciones sobre el estilo de uno o más elementos (W3C, 2014).

#### 4.3.1.3 JAVASCRIPT

JavaScript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsa botones y ventanas con mensajes de aviso al usuario.

Técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos en JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (Eguiluz Montero, 2013).

#### 4.3.1.4 PYTHON

Python es un lenguaje interpretado<sup>18</sup>, interactivo y orientado a objetos que ofrece una gran cantidad de estructuras de datos de alto nivel por medio de un tipado dinámico y fuerte. Es multiparadigma y multiplataforma.

En Python el código fuente se traduce a un pseudo código máquina intermedio llamado bytecode la primera vez que se ejecuta, generando archivos *.pyc* o *.pyo*, que son los que se ejecutarán en sucesivas ocasiones. Python no sólo permite el paradigma de programación imperativo y el orientado a objetos, también puede experimentar programación funcional<sup>19</sup> y programación orientada a aspectos<sup>20</sup>.

---

<sup>18</sup> Un lenguaje de programación interpretado es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora.

<sup>19</sup> La programación funcional es un paradigma de programación declarativa basado en la utilización de funciones aritméticas que no manejan datos de estado.

<sup>20</sup> La programación orientada a aspectos es un paradigma de programación relativamente reciente cuya intención es permitir una adecuada modularización de las aplicaciones y posibilitar una mejor separación de incumbencias.



#### 4.3.1.5 GOOGLE APP ENGINE

Google App Engine (GAE) es una herramienta que abre la infraestructura de producción de Google de forma gratuita como plataforma de desarrollo y hospedaje de aplicaciones web. GAE soporta de manera oficial los lenguajes de programación Python y Java.

GAE posee un entorno de desarrollo denominada SDK de Google App Engine, el mismo que puede ser ejecutado de forma local.

App Engine ejecuta las aplicaciones mediante el intérprete de Python 2.7 cargado en un entorno *sandboxed*<sup>21</sup>. Toda aplicación programada para correr en la nube de Google debe incorporar un archivo de configuración llamado `app.yaml` donde se configuran diferentes aspectos de la aplicación y el entorno de ejecución.

El entorno de ejecución admite módulos de terceros programados íntegramente en Python y no deben incluir ninguna extensión susceptible de ser compilado (Campos, 2011).

#### 4.3.1.6 GOOGLE BIGTABLE

Google BigTable es un motor de bases de datos distribuido, construido por la empresa Google cuyo objetivo principal fue construir un sistema distribuido que acepte datos semi-estructurados para una gran variedad de aplicaciones y dominios.

BigTable almacena información en tablas multidimensionales cuyas celdas están, en su mayoría, sin utilizar. Además, estas celdas disponen de versiones temporales de sus valores, con lo que se puede hacer un seguimiento de los valores que han tomado históricamente (Pérez Costoya, 2012).

### 4.3.2 Instalación y configuración de las herramientas de desarrollo de Course Builder

A continuación se describen los pasos para la implementación del MOOC desarrollado sobre la plataforma Google Course Builder.

#### 4.3.2.1 Instalación de Google App Engine

1. Lo primero que se debe hacer es instalar Python 2.7 en el equipo. Para ello se dirige al siguiente enlace: <http://www.python.org/download/releases/2.7/> y en la sección “Downloads” elegimos la correspondiente a nuestra plataforma, en este caso Windows.
2. Una vez descargado el archivo instalador, se procede a abrirlo. Se abrirá un asistente de instalación en el cuál todas las opciones se dejan las predeterminadas.
3. El siguiente paso consiste en instalar Google App Engine, el mismo que nos permitirá ejecutar Google Course Builder de forma local y también subirlo a la nube para que esté a disposición de todo el público. Para obtener GAE debemos ingresar al siguiente enlace <https://cloud.google.com/appengine/downloads> y descargar *Google App Engine SDK for Python* y escogemos la plataforma Windows.
4. Al igual que con Python, se abre el instalador y se presentará en pantalla el asistente de instalación de GAE en el cuál todas las opciones de configuración quedarán por defecto.

<sup>21</sup> Un entorno *sandboxed*, es aquel entorno de pruebas que aísla los cambios en el código, fruto de la experimentación, del propio entorno de producción o entorno de edición.

5. Cuando la instalación, haya terminado, procedemos a ejecutar la aplicación de Google App Engine. Se obtendrá una ventana igual a la que se puede observar en la figura 4.13

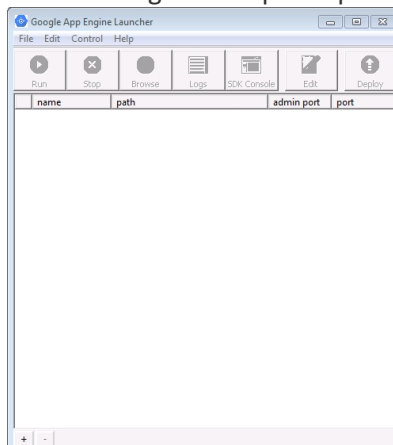


Figura 4.13: Pantalla principal de Google App Engine.

Fuente: Elaboración propia.

6. Como siguiente paso, se procede a descargar Google Course Builder. Para esto, se debe ingresar al siguiente enlace <https://code.google.com/p/course-builder/wiki/DownloadCourseBuilder> en el cual se mostrará una lista de todas las versiones del proyecto de Google disponibles. Para el desarrollo de la presente tesis se utilizó la versión 1.6.0 liberada el 24 de Febrero de 2014.
7. Una vez que la descarga haya terminado se obtiene un archivo .zip el mismo que debe ser descomprimido en cualquier parte del computador que se esté usando.
8. Finalizado el paso anterior, obtenemos la estructura de archivos que se puede observar en la figura 4.14.

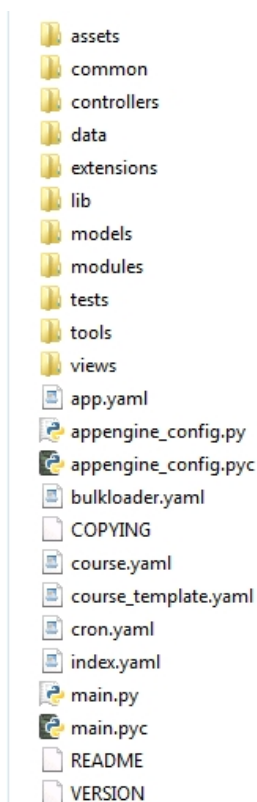


Figura 4.14: Estructura de archivos de Google Course Builder.

Fuente: Elaboración propia.

9. Como siguiente paso, se procede a cargar el proyecto de Google Course Builder en el GAE. Para ello, se ejecuta el GAE instalado en el paso 5, se da clic en el menú *File* y luego en la opción *Add existing application*, lo cual permitirá que se abra la siguiente ventana en pantalla.

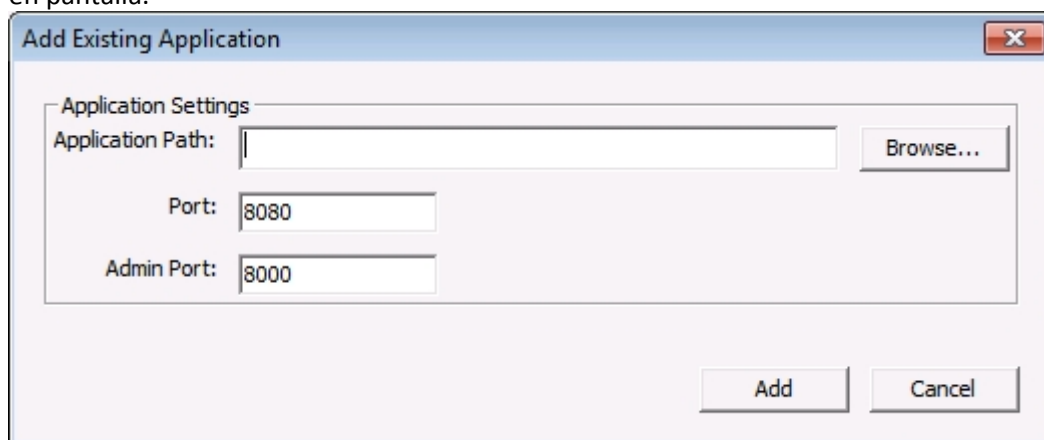


Figura 4.15: Agregar un proyecto de GCB en GAE.

Fuente: Elaboración propia

- Se hace clic en *Browse* y se escoge la carpeta donde se colocaron los archivos en el paso anterior. En la opción *Port* se coloca el puerto por el cual se accederá a la aplicación y en *Admin Port* se coloca el puerto que será utilizado para la administración de Google Course Builder. Para finalizar se debe hacer clic en el botón *Add*.
10. Cuando se tenga ya agregado el proyecto, en la ventana principal del App Engine, se hace clic en el botón *Run* con el cuál se obtiene la aplicación ejecutándose en el servidor local de App Engine.

#### 4.3.2.2 Configuración del archivo `course.yaml`

El archivo `course.yaml` permite personalizar el curso para las necesidades del MOOC que se va a crear. Este archivo se encuentra en la raíz de la carpeta que se descomprimió en el paso 6 de la sección anterior.

En este archivo se va a configurar los siguientes parámetros:

- Correo electrónico del administrador del MOOC
- Nombre del MOOC
- Descripción del curso
- Información del autor (es) del MOOC
- Imagen de la pantalla principal.
- Fecha de inicio.
- URL del foro disponible en el MOOC.
- Idioma del MOOC
- Disponibilidad de curso

Para esta tesis, la configuración del archivo `course.yaml` se puede observar en la figura 4.16.



```

course:
  # A new line or a space separated list of email addresses of course
  # administrative users. Each email address must be placed between '[' and ']',
  # for example: '[test@example.com]'. Regular expressions are not supported,
  # exact match only.
  admin_user_emails: '[programacion.ucuenca@gmail.com]'

  # Course title
  title: 'FUNDAMENTOS DE PROGRAMACION'

  # Main page blurb. (The '|' at the beginning enables multi-line input.)
  blurb: |
    El objetivo de este curso es ayudar a los estudiantes de la materia de Programación de la Facultad de Ingeniería de la Universidad de Cuenca a
    complementar los conocimientos adquiridos de forma presencial, para que logren alcanzar los objetivos de la materia.

  # Information about the instructor to display on the main page.
  instructor_details: 'Andrea Peralta Bravo - Christian Piedra Orellana<br> Tesistas <br> Universidad de Cuenca.'

  main_image:
    url: /assets/img/principal.jpg
    alt_text: Programacion

  start_date: '15 de septiembre de 2014'

  # Forum URL.
  forum_email: fundamentos-de-programacion-ucuenca@googlegroups.com
  forum_url: https://groups.google.com/d/forum/fundamentos-de-programacion-ucuenca

  # Locale for internationalization. See modules/i18n/resources/locale for
  # available locales.
  locale: 'es_ES'

```

Figura 4.16: Configuración del archivo course.yaml.

Fuente: Elaboración propia.

#### 4.3.2.3 Configuración del archivo course\_template.yaml

El archivo course\_template.yaml es una de las nuevas opciones que trae la última versión de Google Course Builder. Permite configurar los datos relacionados con la Institución a la que pertenece el MOOC.

Los parámetros a configurar en este archivo son:

- Nombre de la Institución que ofrece el curso.
- Dirección URL de la Institución
- El logo de la Institución
- Texto a mostrar en la cabecera del MOOC.

```

#####
# Information about your institution
#####

institution:
  # Name of your institution
  name: 'Universidad de Cuenca'

  # URL of your institution's webpage
  url: 'www.ucuenca.edu.ec'

  # Link and alt text for the logo of the course or your institution
  logo:
    url: 'assets/img/your_logo_here.png'
    alt_text: Universidad de Cuenca

  nav_header: 'PROGRAMACION - UNIVERSIDAD DE CUENCA'

```

Figura 4.17: Configuración del archivo course\_template.yaml.

Fuente: Elaboración propia.

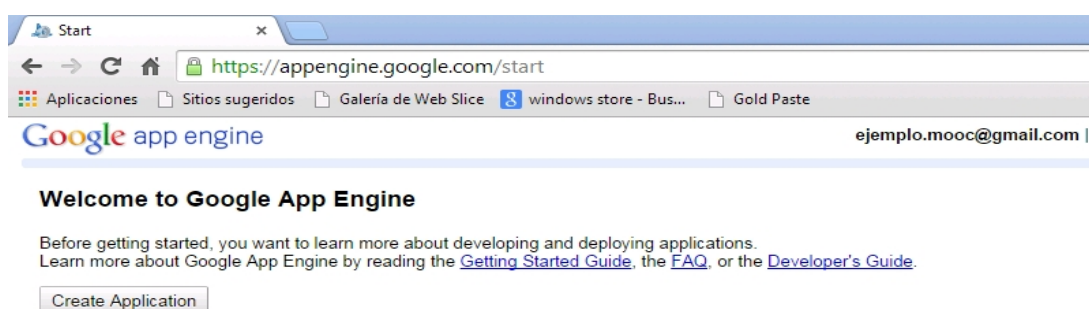
En esta sección cabe mencionar que Google Course Builder acepta cualquier tipo de imágenes. El logo de la Institución debe ser colocado en la carpeta de Google Course Builder dentro de la ruta /assets/img.

#### 4.3.2.4 Creación de aplicación en App Engine.

Una vez que se ha configurado los parámetros del MOOC que se va a crear, el siguiente paso es subirlo a la nube de Google.

Para ello es necesario crear una aplicación en App Engine. Para ello se debe seguir los siguientes pasos:

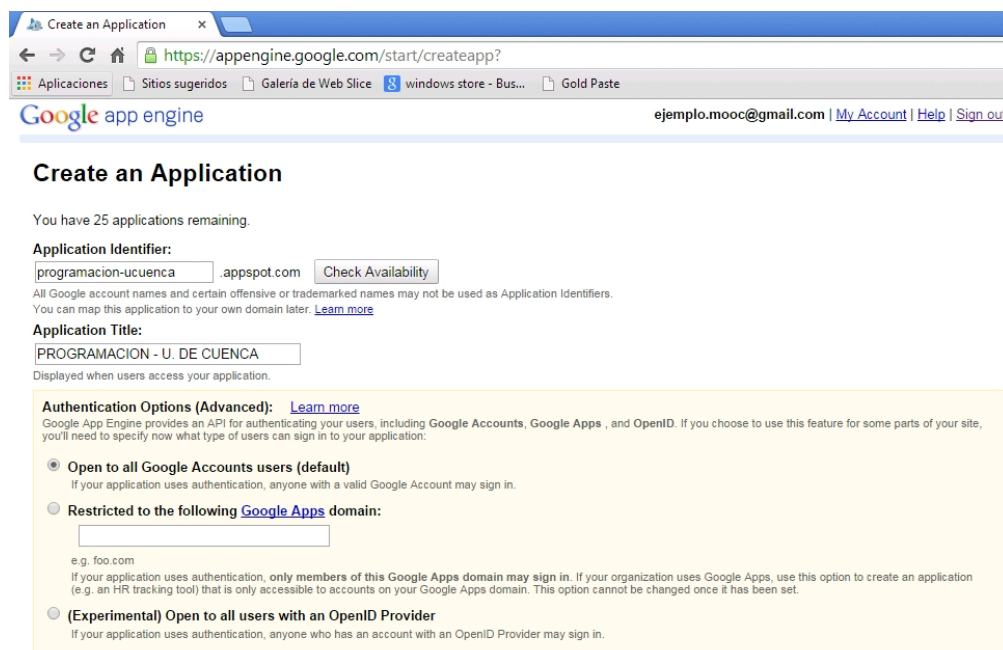
1. Se ingresa al siguiente enlace: <https://appengine.google.com>, se inicia sesión con una cuenta de Gmail y en la ventana principal se hace clic en *Create Application*.



*Figura 4.18: Pantalla principal de App Engine.  
Fuente: Elaboración propia.*

2. Aparecerá una ventana como la que se puede observar en la figura 4.19, en donde se configurarán los siguientes parámetros:
  - a. **Identificador de la aplicación:** Es un identificador que permitirá ingresar al MOOC desde cualquier ordenador que tenga acceso a Internet. Para esta tesis el identificador será *programacion-ucuenca*, por lo que se obtendrá el dominio <https://programacion-ucuenca.appspot.com> con el cual se podrá ingresar al curso.
  - b. **Título de la aplicación:** El nombre del curso que será mostrado en pantalla cuando se ingrese a la aplicación.
  - c. **Opciones de Autenticación:** Permite establecer que cuentas van a poder registrarse en el MOOC. Para ello se dispone de tres opciones:
    - i. Abierto para todos los usuarios con cuentas de Google
    - ii. Restringido para un dominio predeterminado de Google: Está opción permite establecer que únicamente las personas con una cuenta de usuario que use el SMTP de Google puedan registrarse en el MOOC.
    - iii. Abierto para usuarios con un identificador dado por el proveedor: Permite que únicamente las cuentas a las que se el administrador les haya asignado un identificador para registro.

Para esta tesis se usará la primera opción que permitirá que cualquier persona con una cuenta de Google se pueda registrar en el MOOC.



Create an Application

https://appengine.google.com/start/createapp?

Google app engine ejemplo.mooc@gmail.com | My Account | Help | Sign out

### Create an Application

You have 25 applications remaining.

**Application Identifier:**  
 .appspot.com

All Google account names and certain offensive or trademarked names may not be used as Application Identifiers. You can map this application to your own domain later. [Learn more](#)

**Application Title:**

Displayed when users access your application.

**Authentication Options (Advanced):** [Learn more](#)  
 Google App Engine provides an API for authenticating your users, including Google Accounts, Google Apps, and OpenID. If you choose to use this feature for some parts of your site, you'll need to specify now what type of users can sign in to your application:

- ☒ **Open to all Google Accounts users (default)**  
 If your application uses authentication, anyone with a valid Google Account may sign in.
- ☐ **Restricted to the following Google Apps domain:**  
  
 e.g. foo.com  
 If your application uses authentication, only members of this Google Apps domain may sign in. If your organization uses Google Apps, use this option to create an application (e.g. an HR tracking tool) that is only accessible to accounts on your Google Apps domain. This option cannot be changed once it has been set.
- ☐ **(Experimental) Open to all users with an OpenID Provider**  
 If your application uses authentication, anyone who has an account with an OpenID Provider may sign in.

Figura 4.19: Pantalla para crear una aplicación en App Engine.

Fuente: Elaboración propia.

- Finalmente se da clic en *Create Application* y con esto la aplicación quedará registrada para ser usada.

#### 4.3.2.5 Despliegue del MOOC en App Engine

Después de que se ha creado la aplicación que alojará al MOOC en App Engine, se procede a modificar el fichero *app.yaml* que se encuentra en la carpeta en donde se descomprimieron los archivos de Google Course Builder.

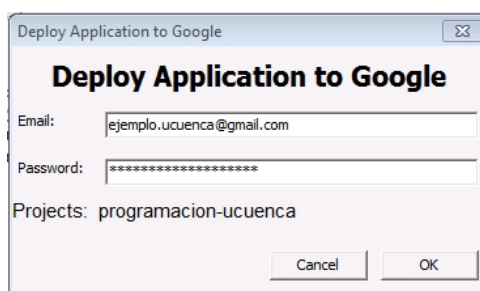
En el archivo mencionado se debe realizar únicamente el cambio en el valor de la línea *application* por el identificador de la aplicación creada en la sección anterior. En la figura 4.20 se observa la configuración de este archivo.

```
1 application: programacion-ucuenca # FIXME: Replace this with your application id
2 version: 1
3 runtime: python27
```

Figura 4.20: Configuración del archivo *app.yaml*

Fuente: Elaboración propia

Una vez realizado este cambio, se ejecuta el SDK de App Engine instalado anteriormente e ingresamos a la opción *Deploy*, e ingresamos el correo electrónico y contraseña con la que se creó la aplicación en el App Engine.



Deploy Application to Google

### Deploy Application to Google

Email:

Password:

Projects:

Figura 4.21: Configuración del despliegue de Google Course Builder.

Fuente: Elaboración propia

De esta manera ya se puede ingresar al MOOC con el dominio obtenido en la sección anterior.

#### 4.3.2.6 Configuración del MOOC desde la nube

Luego que se encuentra desplegada el MOOC en la nube, se procede a ingresar a <https://programacion-ucuenca.appspot.com> e iniciar sesión con la cuenta con la que se creó la aplicación en App Engine y se obtiene un menú como el que se muestra en la figura 4.22.



Figura 4.22: Menú principal del MOOC para la cuenta de administrador.

Fuente: Elaboración propia

Para continuar con la implementación del MOOC, se ingresa al menú *Dashboard* ubicado a la izquierda del menú. La pantalla que se mostrará corresponde a la indicada en la figura 4.23.

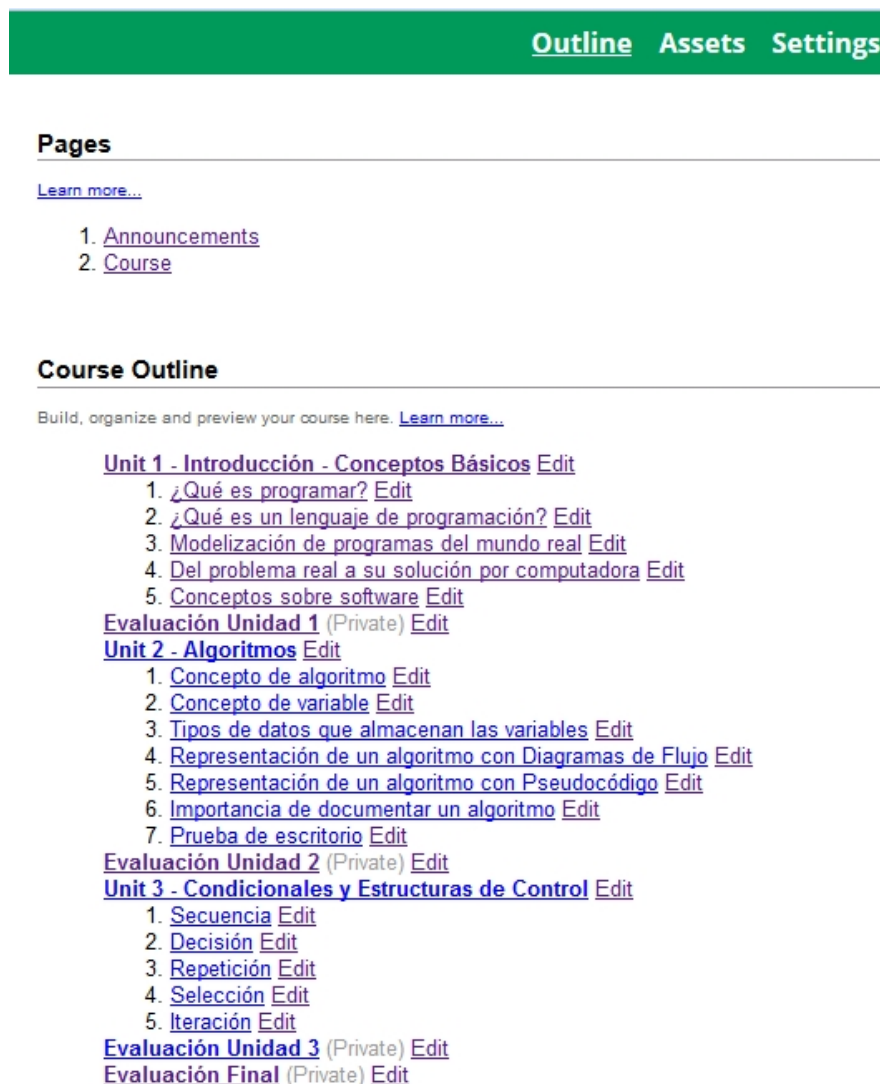


Figura 4.23: Dashboard del MOOC.

Fuente: Elaboración propia

Desde esta ventana se procederá a la configuración del MOOC.

#### 4.3.2.7 Creación de Unidades

Las unidades en Google Course Builder tienen la siguiente estructura:

- **Id:** Identificador de la Unidad.
- **type:** Puede tener los siguientes valores:
  - A: actividad
  - U: unidad
  - O: Otros (link, videos, etc.)
- **Unit\_id:** Su valor depende del campo anterior.
  - A: sufijo del fichero de JavaScript que contiene la actividad.
  - U: Identificador de la unidad.
  - O: link del material externo
- **title:** Título de la unidad.
- **release\_date:** Fecha a partir de la cual está disponible la unidad. No es un campo obligatorio.
- **now\_available:** Si la unidad está disponible o no para ser cursada.

Los datos id, type y unit\_id son configurados automáticamente por la aplicación. El resto de campos se los debe ingresar desde el menú *Add Unit*. En la figura 4.24 se puede observar la configuración de la Unidad 1 del MOOC.



Figura 4.24: Configuración de unidades.  
Fuente: Elaboración propia.

#### 4.3.2.8 Creación de Lecciones

Las lecciones en Google Course Builder tienen la siguiente estructura:

- **ID:** Identificador de la lección.
- **Title:** Título de la unidad a la que pertenece la lección
- **Parent Unit:** Identificador de la unidad a la que pertenece.
- **Video ID:** Identificador de YouTube del video de la lección.
- **Scored:** Establece si las actividades propuestas en la lección van a ser consideradas para la evaluación del curso. Para este caso se elegirá la opción *Questions only give a feedback* (Las preguntas sólo ofrecerán retroalimentación) tal cual se estableció en el diseño del curso.
- **Lesson Body:** Contiene la parte introductoria de la unidad. Utiliza código HTML en la cual se establece el contenido de la lección.
- **Notes:** URL de la *versión en texto* de la lección.

En caso de que la lección tenga actividades:

- **Activity Title:** Título de la actividad

- En la figura 4.25 se puede observar la configuración de la lección 1 perteneciente a la Unidad 1 del MQOC.

ID 3

Title

Parent Unit

Video ID

Provide a YouTube video ID to embed a video.

Scored

Whether questions in this lesson will be scored (summative) or only provide actual feedback (formative).

Lesson Body

```
<span style="font-weight: bold;">Informática: &nbsp;&nbsp;&nbsp;</span><span style="text-indent: -1&nbsp;pt; line-height: 1.22;" lang="ES">Es la ciencia que estudia el análisis y resolución de problemas utilizando computadoras.<br><br><span style="font-weight: bold;">Computadora:&nbsp;&nbsp;&nbsp;</span><span style="text-indent: -1&nbsp;pt; line-height: 1.22; font-family: Symbol;" lang="ES"><span style="font-size: 7pt; line-height: normal; font-family: 'Times New Roman';">&nbsp;&nbsp;&nbsp;</span></span><span style="text-indent: -1&nbsp;pt; line-height:
```

The lesson body is displayed to students above the video in the default template.

Notes

Provide a URL that points to the notes for this lesson (if applicable). These notes can be accessed by clicking on the "Text Version" button on the lesson page.

Activity Title

This appears above your activity.

Activity Listed ☒

Whether the activity should be viewable as a stand-alone item in the unit index.

Activity

Create an activity by entering the correct syntax above. [Learn more...](#)

[Import to lesson body...](#)

---

Status

Andrea Carolina Peralta Bravo  
Christian Patricio Piedra Orellana



La figura 4.25 nos muestra además que el *lesson\_body* está en HTML. Google Course Builder facilita un editor de texto básico para definir este parámetro, pero si se desea centrar imágenes, insertar tablas, insertar formularios, se lo debe realizar insertando código HTML. El desarrollo de las actividades será el tema de la siguiente sección.

### 4.3.2.9 Creación de actividades

Dentro de Google Course Builder, las actividades están desarrolladas en JavaScript. La estructura de una actividad es la siguiente:

- Se crea una variable *activity*, de la siguiente manera: `var activity = [...]`; todo lo que va dentro de los corchetes será el contenido de la actividad.
- Luego, se crea la pregunta la misma que debe ser escrita en HTML para que lo reconozca el navegador.
- Finalmente agregamos las opciones de respuesta para que el estudiante conteste la pregunta. En la figura 4.26 se observa la estructura de una de las actividades creadas para este MOOC.

```
var activity = [
  '<p><b>1.</b> Se conoce como software al equipamiento lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos nece',
  {
    questionType: 'multiple choice',
    choices: [
      ['a', 'Verdadero',true,'a) ¡Correcta! Felicidades.'],
      ['b', 'Falso',false,'b) Respuesta incorrecta. Recuerda que: Se conoce como software al equipamiento lógico o soporte lógico de un sistema informáti']
    ]
  },
  '<p><b>2.</b>Ordene: <br/>Las etapas en la resolución de problemas por computadora son:<br/><ol><li>Análisis del problema</li><li>Especificación de algoritmos</li><li>Diseño</li><li>Escritura</li><li>Ejecución</li></ol>',
  {
    questionType: 'multiple choice',
    choices: [
      ['a', '1,2,3,4,5',false,'a) Respuesta incorrecta. Recuerda que: El orden es: Análisis del problema, Diseño, Especificación de algoritmos, Escritura'],
      ['b', '1,4,2,5,3',true,'b) ¡Correcta! Felicidades.'],
      ['c', '1,2,4,3,5',false,'c) Respuesta incorrecta. Recuerda que: El orden es: Análisis del problema, Diseño, Especificación de algoritmos, Escritura'],
      ['d', '4,1,3,2,5',false,'d) Respuesta incorrecta. Recuerda que: El orden es: Análisis del problema, Diseño, Especificación de algoritmos, Escritura']
    ]
  },
];
```

Figura 4.26: Estructura de actividad en Course Builder.

Fuente: Elaboración propia.

En esta sección se debe tomar en cuenta algunas consideraciones:

- Todo el texto que no es propio de JavaScript debe ir entre comillas simple ( ' ') y escrito en HTML.
- Cada una de las debe ir entre llaves ({...}) y terminar con una coma (,).
- El grupo de opciones va entre corchetes ([...]).
- Cada una de las opciones va entre corchetes ([...]) y terminan con una coma (,), exceptuando la última.
- Las opciones cuentan con tres parámetros:
  - La opción que se mostrará en pantalla al estudiante.
  - True o false para establecer si la pregunta es la respuesta correcta o no.
  - La retroalimentación en caso de que la respuesta sea incorrecta o el mensaje de motivación para continuar con el MOOC en caso de que la respuesta sea correcta.
- La variable creada termina con punto y coma (;)



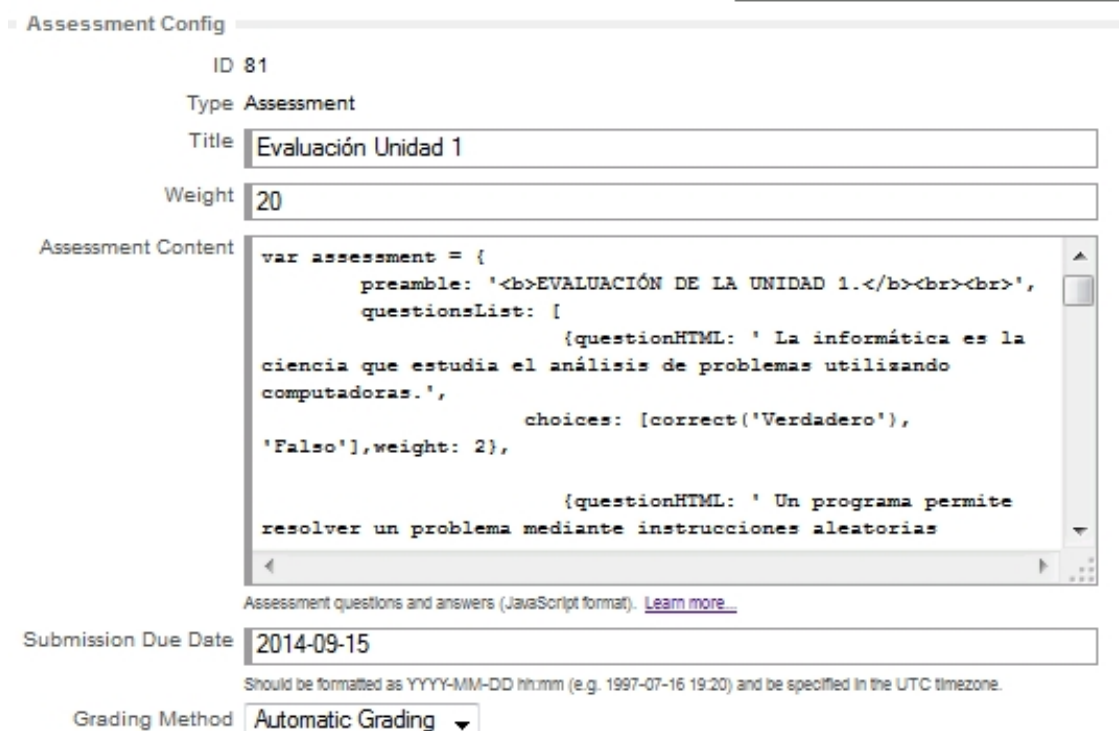
Para crear las actividades se recomienda utilizar un editor de archivos JavaScript (Netbeans, Eclipse, Notepad++, etc.) ya que en la página de configuración de Course Builder se debe colocar el código en un cuadro de texto y no permite visualizar si el código escrito tiene errores.

#### 4.3.2.10 Creación de evaluaciones

La creación de la evaluaciones es muy similar a la creación de actividades revisado en la sección anterior.

Para crear una evaluación se debe seguir los siguientes pasos:

1. Dentro de la página principal del dashboard, se hace clic en *Add Assessment*.
2. Procedemos a configurar los parámetros de la evaluación.
  - a. **Title:** Título de la evaluación
  - b. **Weight:** El peso que tendrá esta evaluación para el curso. Para esta tesis el peso sería 20 para las evaluaciones de unidad y 40 para la evaluación final.
  - c. **Assessment Content:** Contenido de la evaluación escrito en JavaScript.
  - d. **Submission Date:** Fecha en la que se debe dar el examen.
  - e. **Grading Method:** Se escoge el método de calificación de la evaluación. Para nuestro caso se elegirá *Automatic Grading* que significa que una vez que el estudiante haya finalizado la evaluación la plataforma se encargará de validar las respuestas y mostrar la calificación obtenida.



**Assessment Config**

ID 81

Type Assessment

Title Evaluación Unidad 1

Weight 20

Assessment Content

```
var assessment = {
  preamble: '<b>EVALUACIÓN DE LA UNIDAD 1.</b><br><br>',
  questionsList: [
    {questionHTML: ' La informática es la
ciencia que estudia el análisis de problemas utilizando
computadoras.',
      choices: [correct('Verdadero'),
'Falso'],weight: 2},
    {questionHTML: ' Un programa permite
resolver un problema mediante instrucciones aleatorias
```

Assessment questions and answers (JavaScript format). [Learn more...](#)

Submission Due Date 2014-09-15

Should be formatted as YYYY-MM-DD hh:mm (e.g. 1997-07-16 19:20) and be specified in the UTC timezone.

Grading Method Automatic Grading

Figura 4.27: Configuración de evaluaciones en Google Course Builder.

Fuente: Elaboración propia.

La opción *Assessment Content* tiene una estructura similar a la de las actividades con algunas diferencias:

- El nombre de la variable que se crea es *assessment*.
- La cabecera de la evaluación se la debe colocar luego de la palabra reservada *preamble*, entre comilla simple ('...') y en HTML.
- Todas las preguntas deben ir dentro de la opción *questionList* entre corchetes ([...]).

- Cada pregunta debe ir dentro de llaves ({...}).
- La pregunta se la coloca después de *questionHTML* entre comillas, en HTML y seguida de una coma (,).
- Las opciones de respuesta se deben colocar luego de *choices* y la opción que sea correcta va precedida de la palabra *correct* y entre paréntesis.
- La opción *weight* se la coloca luego de las opciones de respuesta y sirve para determinar el peso de cada pregunta en la evaluación.

En la figura 4.28 se muestra la estructura de la primera evaluación desarrollada para esta tesis.

```
var assessment = {
  preamble: '<b>EVALUACIÓN DE LA UNIDAD 1.</b><br><br>',
  questionsList: [
    {questionHTML: ' La informática es la ciencia que estudia el análisis de problemas utilizando computadoras.',
      choices: [correct('Verdadero'), 'Falso'],weight: 2},
    {questionHTML: ' Un programa permite resolver un problema mediante instrucciones aleatorias ejecutadas en el ordenador.',
      choices: ['Verdadero', correct('Falso')],weight: 2},
    {questionHTML: ' Un algoritmo es:',
      choices: [correct('Secuencia de pasos e instrucciones que producen un resultado deseado y es finito'),'Secuencia de pasos e instrucciones c'],
    {questionHTML: ' Las etapas en la resolución de un problema son:',
      choices: ['Análisis, Especificación, Escritura, Verificación','Análisis, Diseño, Especificación, Escritura', 'Análisis, Implementación, Escr'],
    {questionHTML: ' La descomposición en módulos no es una herramienta útil para resolver un problema.',
      choices: ['Verdadero', correct('Falso')],weight: 2},
    {questionHTML: ' Un programa es un conjunto de instrucciones. Cuando se escribe por ejemplo las directrices para llegar a la casa de un amigo, se est',
      choices: [correct('Verdadero'), 'Falso'],weight: 2},
    {questionHTML: ' Una especificación es el proceso de analizar los problemas del mundo real y determinar de forma clara y concreta un objetivo.',
      choices: [correct('Verdadero'), 'Falso'],weight: 2},
    {questionHTML: ' Python:',
      choices: ['Es un lenguaje de programación interpretado, con tipado dinámico sólo para plataformas Windows.','Es un lenguaje de programación c'],
    {questionHTML: ' Especificar un problema significa establecer en forma unívoca su contexto, precondiciones y el resultado esperado.',
      choices: [correct('Verdadero'), 'Falso'],weight: 2},
    {questionHTML: ' A partir de un problema se debe realizar un proceso de abstracción.',
      choices: [correct('Verdadero'), 'Falso'],weight: 2},
  ],
  checkAnswers: false
}
```

Figura 4.28: Estructura de evaluación en Course Builder.

Fuente Elaboración propia.

#### 4.3.2.11 Modificar la presentación del MOOC.

Para cambiar la parte visual de nuestro MOOC existen dos archivos que se pueden modificar: *butterbar.css* y *main.css*, los mismos a los que puede acceder desde el dashboard del curso en el menú *Assets*.

Para el desarrollo de la presente tesis, estos dos archivos fueron utilizados para que el MOOC tenga los colores de la Universidad de Cuenca.

En la figura 4.29 y 4.30, se puede observar parte del código fuente de estos archivos una vez que han sido modificados para las necesidades de este proyecto.

```

.gcb-butterbar-dashboard {
  margin-top: 5px;
}
.gcb-butterbar.shown {
  opacity: 1;
  visibility: visible;
  left: 0;
}
.gcb-butterbar a,
.gcb-butterbar p {
  margin-bottom: 0;
  margin-top: 0;
  line-height: 29px;
  font-size: 11px;
  display: inline;
}
.gcb-butterbar a {
  color: #424027;
  text-decoration: underline;
  z-index: 1002;
  position: relative;
}
#gcb-butterbar-close {
  color: #888888;
  text-decoration: none;
  font-weight: bold;
  margin-left: 5px;
  font-size: 13px;
  cursor: default;
}
#gcb-butterbar-close:hover {
  color: #555555;
}

```

Figura 4.29: Archivo butterbar.css.  
Fuente: Elaboración Propia

```

/* Styles for the course title shown at the top of all student-facing pages. */
.gcb-title-header {
  border: solid #babcc0;
  border-width: 0 0 0 1px;
  float: left;
  font-family: "open sans", arial, sans-serif;
  font-size: 27px;
  font-weight: 300;
  line-height: 1.52;
  margin: 0;
  padding: 0 12px;
  text-shadow: 0 1px 1px #fff;
  white-space: nowrap;
}

/* Styles for the course logo at the top-left of all student-facing pages. */
.gcb-course-image {
  float: left;
  font-size: 27px;
  font-weight: 300;
  line-height: 1.52;
  margin: 0;
  padding: 0 12px 0 0;
  text-shadow: 0 1px 1px #fff;
  white-space: nowrap;
}

/* Styles for course summary headers. */
.gcb-product-headers-large {
  color: #dd4b39;
  font-family: "open sans", arial, sans-serif;
  font-size: 25px;
  font-weight: normal;
  margin: 10px 7px;
}

```

Figura 4.30: Archivo main.css.  
Fuente: Elaboración propia

#### 4.3.2.12 Cambiar las opciones de configuración del curso

La opción *Settings* dentro del dashboard del MOOC que se está creando permite modificar algunas de los parámetros del curso que se realizaron al inicio de la implementación. Estos parámetros son:

- Nombre del Curso
- Correo electrónico del administrador
- E-mail del foro
- E-mail de anuncios
- El idioma del curso
- Fecha de inicio
- Si el curso está disponible
- Si personas que no están registradas pueden acceder a los contenidos del curso, etc.

En la figura 4.31 y 4.32 se observa las configuraciones realizadas en esta opción para el curso.

— Basic Course Settings —

▣ Course Config

Course Name

Course Admin Emails   
A space-separated list of email addresses of course administrators. Each email address must be placed between '[' and ']'.

Forum Email   
Email for the forum, e.g. 'My-Course@googlegroups.com'.

Announcement List Email   
Email for the mailing list where students can register to receive course announcements, e.g. 'My-Course-Announce@googlegroups.com'

Locale

Course Start Date

Make Course Available ☒

Make Course Browseable ☐  
Allow non-registered users to view course content.

ID for Google Analytics   
This ID tells Google Analytics who is calling, and allows it to string together routes that visitors take through pages. Obtain this ID by signing up at <http://www.google.com/analytics>

ID for Google Tag Manager   
This ID tells Google Tag Manager who is calling. This allows the Tag Manager to notify other site use tracking services what users are doing on the site. Obtain this ID by signing up at <http://www.google.com/tagmanager>

▣ Unit and Lesson Settings

Hide Lesson Navigation Buttons ☐  
Whether to hide the 'Previous Page' and 'Next Page' buttons below lesson and activity pages

Figura 4.31: Configuración del curso desde el dashboard.  
 Fuente. Elaboración Propia.

Enable Registrations ☒

### Additional Fields

**Rich Text**

```
<b><br>2. Apellidos:<input type="text" size="30" /><br>3. Edad:<input type="text" size="30" /><br>4. Sexo:<input type="radio" value="Femenino" /> Femenino<br><input type="radio" value="Masculino" /> Masculino<br>5. Ciudad de origen: <input type="text" size="30" /><br>6. ¿Ha tomado antes un curso o materia de programación:<input type="radio" value="si" /> Si<br><input type="radio" value="no" /> No
```

### Instructor Details

**Rich Text**

### Course Abstract

**File** **Text**

El objetivo de este curso es ayudar a los estudiantes de la materia de Programación de la Facultad de Ingeniería de la Universidad de Cuenca a complementar los conocimientos adquiridos de forma presencial, para que logren alcanzar los objetivos de la materia.

Course Video

Course Image

/assets/img/logoCurso.png

URL for the preview image shown on the course homepage. This will only be shown if no course video

Fuente. Elaboración Propia

La configuración de este campo se lo revisará en la siguiente sección.



Como se mencionó anteriormente, dentro de la opción *Settings* del dashboard, se puede crear el cuestionario de registro de estudiantes en el MOOC.

En la figura 4.33 se puede observar el código utilizado para esta sección.

```
<div><br>2. Apellidos:  <input name="edad" size="30" type="text">  
<br>3. Edad:  <input name="edad" size="30" type="text">  
<br>4. Sexo:  <input name="sexo" value="Femenino" type="radio"> Femenino  <input name="sexo" value="Masculino" type="radio"> Masculino  
<br>5. Ciudad de origen:  <input name="ciudad" size="30" type="text">  
<br>6. ¿Ha tomado antes un curso o materia de programación?  <input name="curso_antes" value="si" type="radio"> Si  <input name="curso_antes" value="no" type="radio"> No  
-En caso de que su respuesta sea afirmativa, ¿Dónde?  <input name="curso_antes" value="Colegio" type="radio"> Colegio  
<input name="curso_antes" value="Institutos" type="radio"> Institutos  Otros (Especifique):  
<br>7. Programa en algún lenguaje de programación:  <input name="prog_antes" value="si" type="radio"> Si  <input name="prog_antes" value="no" type="radio"> No  
-En caso de que su respuesta sea afirmativa, ¿Cuál?  <input name="cuales" size="30" type="text">  
</p>  
<p>  
<br>8. Es repetidor de la asignatura?  <input name="repetidor" value="si" type="radio"> Si  <input name="repetidor" value="no" type="radio"> No  
<br>9. Vaya al siguiente enlace:  
<a href="http://www.estilosdeaprendizaje.es/chaea/chaea.htm" target="_blank">enlace</a>  
<a href="http://www.estilosdeaprendizaje.es/chaea/chaea.htm" target="blank">http://www.estilosdeaprendizaje.es/chaea/chaea.htm</a>  
<br>10. Conteste el cuestionario y anote los resultados para cada uno de los siguientes valores.  
<br>Activo:  <input name="estilo" size="2" type="text">  
Reflexivo:  <input name="estilo" size="2" type="text">  
Pragmático:  <input name="estilo" size="2" type="text">  
Teórico:  <input name="estilo" size="2" type="text">
```

*Figura 4.33: Código para la creación de registro de estudiantes.*  
Fuente: Elaboración propia

Para la creación de anuncios en el MOOC, se debe ir al dashboard y en la sección *Outline* ingresar a *Announcements*.

Se abrirá una ventana que contiene todos los avisos del curso, se ingresa en *Add New* y se configuran los siguientes parámetros:

- **Title:** Título del aviso.
- **Date:** Fecha de publicación del aviso.
- **Body:** Contenido del anuncio. Debe estar en HTML.
- **Status:** Si es un borrador o si se publicará el aviso cuando se guarden los cambios realizados.
- **Send e-mail:** Permite decidir si el aviso será enviado o no a los estudiantes del MOOC por correo electrónico.

## — Announcement —

ID ahZzfnByb2dyYW1hY2lvbi11Y3VlbnNhch8LEhJBbm5vdW5jZW1lbnRFbnRpdHkYglCAglCAgA

Title Bienvenidos al Curso

Date 2014/09/15

Body

Rich Text

<br>Estimados estudiantes, bienvenidos al curso  
 ALGORITMOS, DATOS Y ESTRUCTURAS.<br><br>Aquí se  
 abordaran los tres primeros capítulos del sílabo de la asignatura:  
 <br><ul><li>Capítulo 1: Introducción - Conceptos Básicos</li>  
 <li>Capítulo 2: Algoritmos</li><li>Capítulo 3: Condicionales y  
 Estructuras de Control.<br></li></ul>Le recordamos que tendrá  
 una duración de 10 horas distribuidos en dos semanas, y estará  
 disponible del 15 de septiembre al 26 de septiembre del

Status Published

Send Email ☒

Email will be sent to : anuncios-mooc-ucuenca@googlegroups.com

Save

Close

Figura 4.34: Configuración de avisos para el MOOC.

Fuente: Elaboración propia

**4.3.2.15 Obtención de datos de los estudiantes registrados en el MOOC**

Para la obtención de estos datos en Google Course Builder recomiendan crear un script con las rutas en donde se encuentran librerías necesarias para la descarga:

```
import sys
sys.path.insert(0, 'C:/coursebuilder')
sys.path.insert(0, 'C:/Program Files (x86)/Google/google_appengine')
sys.path.insert(0, 'C:/Program Files (x86)/Google/google_appengine/lib/fancy_urllib')
sys.path.insert(0, 'C:/Program Files (x86)/Google/google_appengine/lib/jinja2-2.6')
sys.path.insert(0, 'C:/Program Files (x86)/Google/google_appengine/lib/webapp2-2.5.2')
sys.path.insert(0, 'C:/Program Files (x86)/Google/google_appengine/lib/webob-1.2.3')
sys.path.insert(0, 'C:/Program Files (x86)/Google/google_appengine/lib/yaml/lib')

import argparse
import etl
if __name__ == '__main__':
    PARSER = argparse.ArgumentParser(add_help=False, parents=[etl.PARSER])
    etl.main(PARSER.parse_args())
```

Figura 4.35: Datos de Script para descarga de información.

Fuente: Elaboración propia

El comando que se debe ejecutar para poder descargar la información es el siguiente:

```
python tools/etl/gcbpaths.py download datastore <course_url_prefix> <app_name>
<app_server_name> --archive_path <archive_path> --datastore_types
<model1,model2>
```



En donde:

**<course\_url\_prefix>**: es el URL en donde se alojan los datos que queremos obtener.  
**<app\_name>**: es el nombre que tiene la aplicación en el servidor.  
**<app\_server\_name>**: es el nombre del servidor en donde se aloja el MOOC.  
**<archive\_path>**: es la ruta en donde se almacenaran los datos que se van a descargar.  
**<model1,model2>**: son el tipo de datos que se quiere descargar entre estos Student, StudentAnswersEntity, StudentPropertyEntity.

A continuación se muestra la ejecución del comando:

```
C:\Users\Public\Documents\trabajos de ciclo\tesis\Google Course Builder>python tools\etl\gcbpaths.py download datastore / programacion-ucuenca programacion-ucuenca.appspot.com --archive_path ./coursedataST.zip --datastore_types Student
```

Figura 4.36: Comando para descarga de datos del MOOC.

Fuente: Elaboración propia

Para poder completar la descarga es necesario poner el Usuario y la Contraseña de administrador, entonces lo que se descarga es un archivo de tipo JSON con los datos de los estudiantes.

## 4.4 Recapitulación

En el presente capítulo se ha realizado el diseño y la implementación del MOOC que será la base fundamental del siguiente paso es la construcción del modelo conceptual. Se ha definido una duración de 10 horas para el mismo.

El puntaje del MOOC está dividido en 3 evaluaciones de 20 puntos cada una, cada evaluación corresponde a un módulo del MOOC y una evaluación de los 3 capítulos de la asignatura sobre un total de 40 puntos. Al finalizar el MOOC, el estudiante deberá rendir una prueba global sobre 40 puntos, con la cual se definirá el puntaje obtenido en el MOOC.

En cuanto al diseño, la guía desarrollada por la Universidad Carlos III de Madrid fue de gran utilidad puesto que utiliza una metodología sencilla para la creación del MOOC pero que sirve como base fundamental al momento de la implementación para crear una estructura del MOOC que permita alcanzar los objetivos propuestos antes de iniciar el curso.

Se ha creado una cuenta de Google, con el fin de generar el foro, un repositorio en Google Drive para almacenar todo el material necesario para el MOOC (textos, informes, etc.) y un canal de YouTube en el cual estarán alojados todos los videos del MOOC. Esto es de gran utilidad puesto que al ser Google Course Builder la plataforma escogida para la implementación del MOOC, esta permite la fácil interacción con las distintas aplicaciones de Google.



# **CAPÍTULO 5:**

## **Diseño e Implementación de un Data Warehouse para análisis de resultados del MOOC**

### **5.1 Introducción**

En el presente capítulo se diseñó e implementó un Data Warehouse, que se utilizó en un caso de estudio que permitió analizar los resultados obtenidos a través de la puesta en marcha del MOOC creado en el Capítulo 4.

### **5.2 Fundamentación Teórica**

En esta parte se introduce el concepto de Data Warehouse y se describen metodologías que permiten su desarrollo.

#### **5.2.1 Data Warehouse**

Un Data Warehouse es una base de datos corporativa que se caracteriza por integrar y depurar la información de una o más fuentes distintas, para luego procesarla permitiendo su análisis desde varias perspectivas y con grandes velocidades de respuestas (Sinnexus, 2013).

El término Data Warehouse fue utilizado por primera vez por Bill Inmon<sup>22</sup> y se traduce al español como *almacén de datos*. Según Inmon, un Data Warehouse se caracteriza tener las siguientes características:

- **Integrado:** los datos almacenados en el Data Warehouse deben integrarse en una estructura consistente, por lo que las inconsistencias existentes entre los diversos sistemas operacionales deben ser eliminadas. La información suele estructurarse también en distintos niveles de detalle para adecuarse a las distintas necesidades de los usuarios.
- **Temático:** Sólo los datos necesarios para el proceso de generación del conocimiento del negocio se integran desde el entorno operacional. Los datos se organizan por temas para facilitar su acceso y entendimiento por parte de los usuarios finales.
- **Histórico:** El tiempo es parte implícita de la información contenida en un Data Warehouse. En los sistemas operacionales, los datos siempre reflejan el estado de la actividad del negocio en el momento presente. Por el contrario, la información almacenada en el Data Warehouse sirve para realizar análisis de tendencias. Por lo tanto, el Data Warehouse se carga con los distintos valores que toma una variable en el tiempo para permitir comparaciones.
- **No volátil:** Los datos almacenados en un Data Warehouse existen para ser leídos pero no modificados. La información es por lo tanto permanente, significando la actualización del Data Warehouse la incorporación de los últimos valores que tomaron las distintas variables contenidas en él sin ningún tipo de acción sobre lo que ya existía (Sinnexus, 2013).

Un Data Warehouse tiene los siguientes procesos que son claves para su construcción:

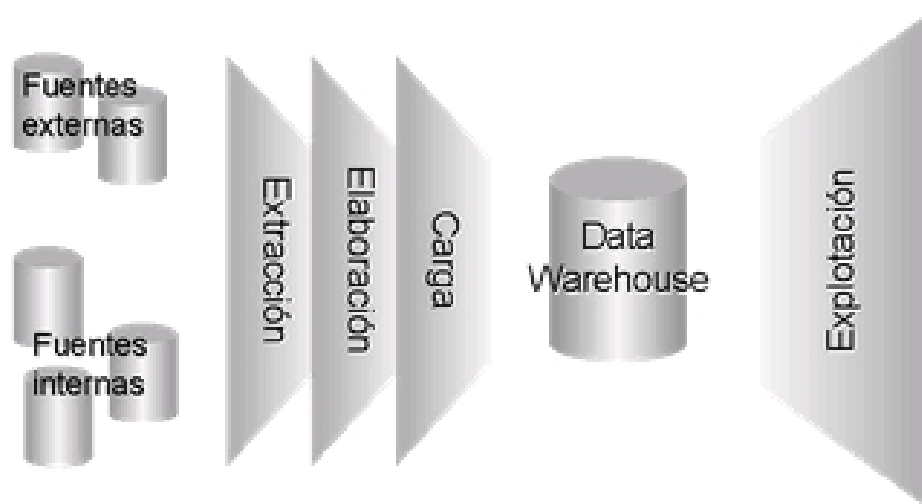


Figura 5.1: Procesos del Data Warehouse.

Fuente: (Fernández, 2012)

- **Extracción:** Obtención de información de las distintas fuentes tanto internas como externas.
- **Elaboración:** Filtrado, limpieza, depuración, homogeneización y agrupación de la información.

<sup>22</sup> Bill Inmon, conocido como el padre del Data Warehousing, se ha convertido en el autor más prolífico y conocido a nivel mundial en Almacenes de datos e inteligencia de negocios.

- **Carga:** Organización y actualización de los datos y los metadatos en la base de datos.
- **Explotación:** Extracción y análisis de la información en los distintos niveles de agrupación (Fernández, 2012).

## 5.2.2 Modelos multidimensionales

Las bases de datos multidimensionales se utilizan principalmente para crear aplicaciones OLAP (On-Line Analytical Processing) y pueden verse como bases de datos de una sola tabla, su peculiaridad es que por cada dimensión tienen un campo (o columna), u otro campo por cada métrica o hecho, es decir, estas tablas almacenan registros cuyos campos son de la forma:

$$(d_1, d_2, d_3 \dots f_1, f_2, f_3)$$

Donde los campos  $d_i$  hacen referencia a las dimensiones de la tabla y los campos  $f_i$  a las métricas o hechos que se quiere almacenar, estudiar o analizar (Bernabeu, 2010).

Un ejemplo de un modelo multidimensional se lo puede observar en la figura 5.2, la cual consta con una tabla de hechos (Ventas), y tres dimensiones (Cliente, Fecha, Producto)

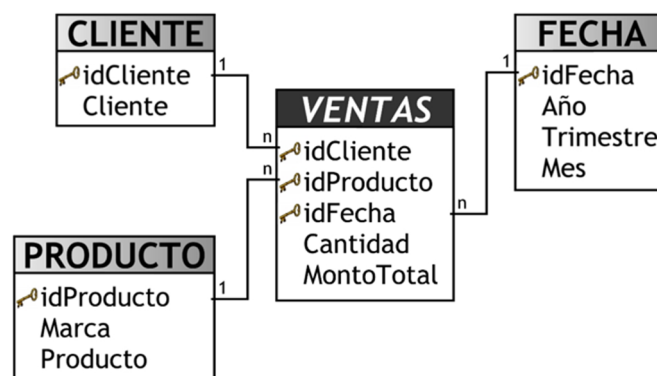


Figura 5.2: Ejemplo de modelo multidimensional.  
Fuente: (Bernabeu, 2010)

## 5.2.3 Descripción de metodologías para Data Warehouse

A continuación, se describen metodologías para el desarrollo de un Data Warehouse:

### 5.2.3.1 Metodología Kimball

La metodología se basa en lo que Kimball denomina ciclo de vida dimensional del negocio. Este ciclo de vida del proyecto de Data Warehouse, está basado en cuatro principios básicos (Rivadera, 2010):

- **Centrarse en el negocio:** consiste en la identificación de los requerimientos del negocio y su valor asociado, y usar estos esfuerzos para desarrollar relaciones sólidas del negocio, agudizando el análisis del mismo y la competencia consultiva de los implementadores.
- **Construir una infraestructura de información adecuada:** diseñar una base de información única, integrada, fácil de usar, de alto rendimiento donde se reflejará la amplia gama de requerimientos de negocio identificados en la empresa.
- **Realizar entregas en incrementos significativos:** crear el almacén de datos en incrementos entregables en plazos de 6 a 12 meses. Se debe usar el valor de negocio de

cada elemento identificado para determinar el orden de aplicación de los incrementos. En esto la metodología se parece a las metodologías ágiles de construcción de software.

- Ofrecer la solución completa: proporcionar todos los elementos necesarios para entregar valor a los usuarios de negocios. Para comenzar, esto significa tener un Data Warehouse sólido, bien diseñado, con calidad probada, y accesible. También se deberá entregar herramientas de consulta ad hoc, aplicaciones para informes y análisis avanzado, capacitación, soporte, sitio web y documentación (Rivadera, 2010).

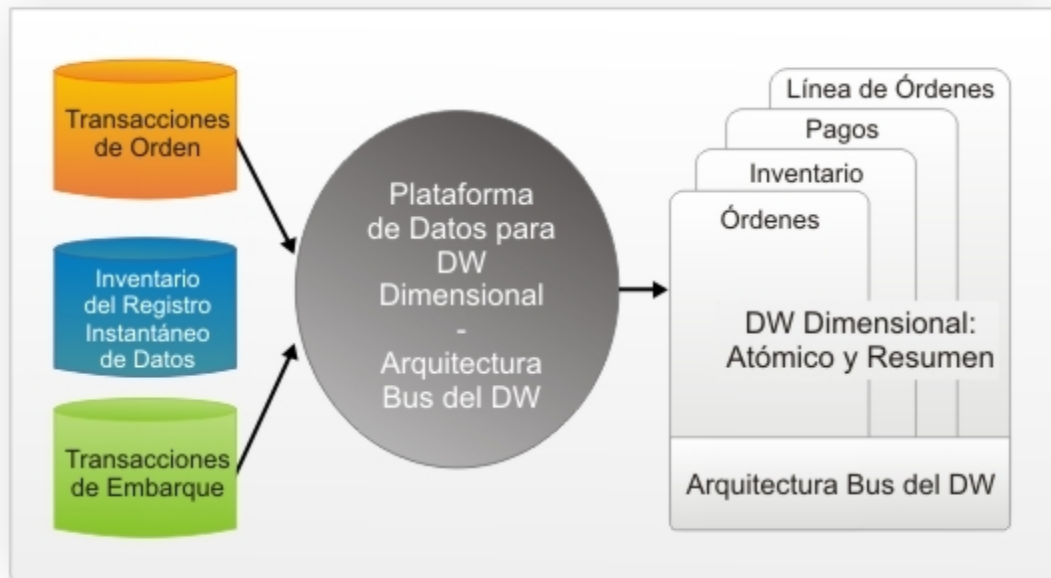


Figura 5.3: Metodología de Kimball.  
Fuente: (Marciales, 2014)

### 5.2.3.2 Metodología CRISP

CRISP-DM (Cross-Industry Standard Process for Data Mining) es una de las principales metodologías para el desarrollo de Data Warehouse y data mining.

La metodología CRISP está sustentada en estándares internacionales que refleja la robustez de sus procesos y que facilitan la unificación de sus fases en una estructura confiable y amigable para el usuario. Además de ello, esta tecnología interrelaciona las diferentes fases del proceso entre sí, de tal manera que se consolida un proceso iterativo y recíproco. Otro aspecto fundamental de esta tecnología es que es planteada como una metodología imparcial o neutra respecto a la herramienta que se utilice para el desarrollo del proyecto de Data Warehouse o Data Mining siendo su distribución libre y gratuita.

El ciclo de vida del proyecto según la metodología CRISP está basado en seis fases cambiantes entre sí y nunca terminantes, lo cual lo postula como un ciclo en constante movimiento (Salcedo Parra, Galeano, & Rodriguez, 2009). En la Figura 5.4, se puede observar el ciclo de vida según la metodología de CRISP.

### 5.2.3.3 Metodología HEFESTO

Es una metodología que combina los objetivos de las metodologías Bill Inmon y Ralph Kimball y que fue creada por Darío Bernabeu. La metodología Hefesto se basa en la recolección de requerimientos que contienen la información del usuario, posterior a ello se continua con los procesos de extracción, transformación y carga de datos (denominado como ETL – *Extract, Transform and Load*) para poder de esta manera definir un esquema lógico que nos permita organizar los datos transformándolos ya sea en Data Marts o Data Warehouse (Bernabeu, 2010).

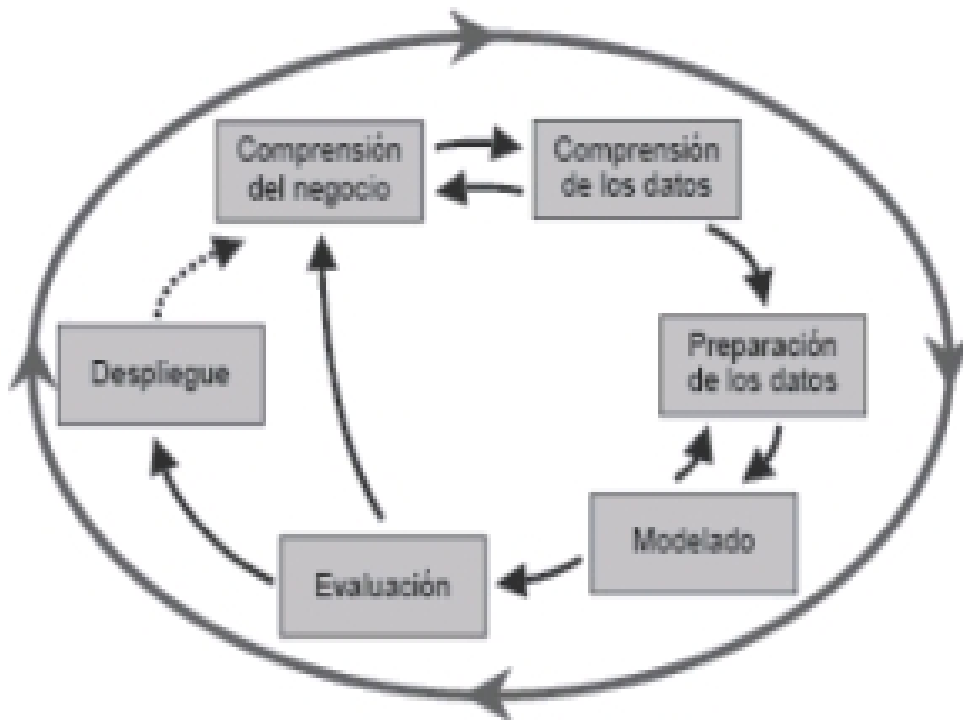


Figura 5.4: Metodología CRISP.  
Fuente: (Goicochea, 2009)

Los pasos de esta metodología son los siguientes:

- Tanto objetivos como resultados esperados que surjan en cada fase son de fácil comprensión y se distinguen rápidamente.
- La estructura de Hefesto tiene la capacidad de adaptarse con facilidad y rapidez a los cambios de negocio debido a su enfoque basado en los requerimientos del usuario.
- Al involucrar a los usuarios finales en cada etapa, permite reducir la resistencia al cambio dado que el usuario puede tomar decisiones respecto al comportamiento y funcionamiento del Data Warehouse.
- Los modelos conceptuales y lógicos que emplea Hefesto son sencillos de interpretar y analizar.
- Tiene la característica de ser independiente del ciclo de vida que se emplea para contener la metodología.
- No tiene dependencia con las herramientas que se empleen para su implementación.
- No tiene dependencia con las estructuras físicas que puedan contener al Data Warehouse así como de su respectiva distribución.
- Es secuencial puesto que cuando culmina con una fase, los resultados que se obtienen de la misma se convierten en el punto de partida para el siguiente paso.

- Puede ser aplicado tanto para Data Warehouse como para Data Mart.

La metodología de Hefesto se basa en cuatro pasos para la construcción de un Data Warehouse, tal como se indica en la Figura 5.5.

A continuación se describen cada uno de los pasos de la metodología HEFESTO (Bernabeu, 2010).

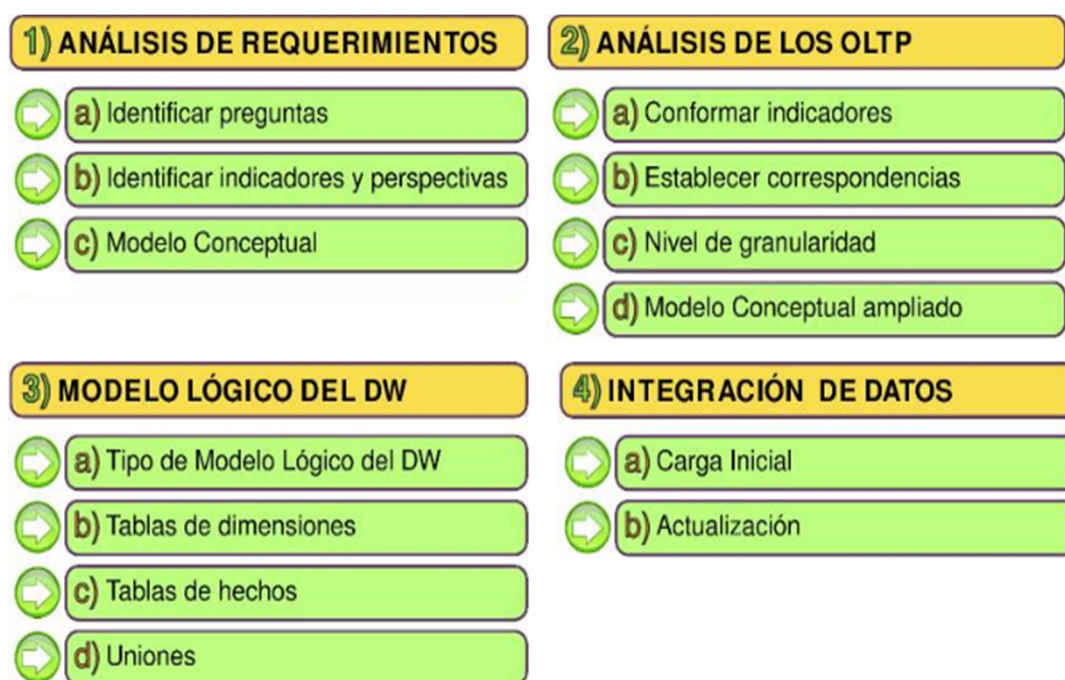


Figura 5.5: Metodología HEFESTO.  
Fuente: (Bernabeu, 2010)

### 1. Análisis de requerimientos

**Identificar preguntas:** Por medio de este punto de partida en el que los usuarios guían el desarrollo puesto que aquí se recolectan todos los requerimientos por medio de la utilización de varias técnicas: entrevistas, cuestionarios, observaciones, etc. Esto permite considerar la identificación de las necesidades de información clave de alto nivel soportada por alguna fuente de información.

**Identificar indicadores y perspectivas:** Una vez realizado el paso anterior se debe descomponer las preguntas de negocio para descubrir los correspondientes indicadores y perspectivas. Los indicadores no son más que valores numéricos que representan lo que se desea analizar. Las perspectivas son objetos mediante los cuales se quiere examinar los indicadores.

**Modelo conceptual:** A partir de los indicadores y perspectivas identificadas se puede ya construir un modelo conceptual que permite comprender los resultados a obtener sin tener conocimientos previos.





## 2. Análisis de los OLTP

**Conformar indicadores:** Por medio de este paso se enseña cómo se calcularían los indicadores definidos anteriormente indicando los hechos que lo componen y la función de sumariación que se utilizaría para la agregación.

**Establecer correspondencias:** En este apartado se van a examinar las fuentes de información disponibles que contengan la información requerida para identificar las correspondencias con el modelo conceptual.

**Nivel de granularidad:** Se realiza la selección de los campos que contendría cada perspectiva, no es necesario considerar todos los datos solo se debe seleccionar únicamente a los más relevantes para las consultas.

**Modelo conceptual ampliado:** Se agrega al modelo conceptual creado en el paso anterior los atributos correspondientes a cada perspectiva.

## 3. Modelo lógico del Data Warehouse

**Tipo de modelo lógico del Data Warehouse:** Se procede a seleccionar el esquema que se adapte mejor a los requerimientos y necesidades del usuario final. Estos nos permiten definir objetivamente si se emplearía un esquema en estrella, constelación o copo de nieve ya que afectaría considerablemente la elaboración del modelo lógico.

**Tablas de dimensiones:** Se procede a diseñar las tablas de dimensión que formarían parte del Data Warehouse, cada perspectiva definida en el modelo conceptual resultaría ser una tabla de dimensión.

**Tablas de hechos:** Se define la correspondiente *tabla de hechos* que contendría los hechos a través de los cuales se constituirían los indicadores.

**Uniones:** Se define la relación entre las tablas de dimensiones y las tablas de hechos.

## 4. Integración de datos

**Carga inicial:** Se procede a poblar el modelo lógico con datos utilizando técnicas de limpieza y calidad de datos, procesos ETL, etc. Primero cargar los datos de las dimensiones y luego los datos de las tablas de hechos.

**Actualización:** Establecer políticas y estrategias de actualización.

# 5.3 Selección de la metodología

En base a la descripción de las características realizado en la sección anterior, se ha creado un cuadro resumen en forma de comparativa entre las tres metodologías mencionadas, el mismo que se puede observar en la tabla 5.1.

METODOLOGÍA	ENFOQUE PRINCIPAL	RESUMEN DE LA METODOLOGÍA
CRISP	Data mining	<ul style="list-style-type: none"> <li>Entendimiento del negocio.</li> <li>Comprensión de los datos</li> <li>Preparación de los datos</li> </ul>

		<ul style="list-style-type: none"> <li>Modelado</li> <li>Evaluación</li> <li>Implementación</li> </ul>
<b>Kimball</b>	Diseño de Data Warehouse	<ul style="list-style-type: none"> <li>Presentación del proyecto</li> <li>Requerimientos del negocio</li> <li>Diseño – Arquitectura técnica.</li> <li>Selección e instalación de herramientas.</li> <li>Modelado dimensional</li> <li>Diseño físico</li> <li>Especificaciones de la herramienta analítica.</li> <li>Desarrollo de la herramienta analítica.</li> <li>Desarrollo del Data Warehouse</li> <li>Mantenimiento del Data Warehouse</li> </ul>
<b>HEFESTO</b>	Construcción del Data Warehouse	<ul style="list-style-type: none"> <li>Análisis de requerimientos</li> <li>Análisis de OLTP.</li> <li>Modelo lógico del Data Warehouse</li> <li>Integración de datos</li> </ul>

Tabla 5.1: Cuadro resumen de las metodologías.

Fuente: Elaboración propia

Para el desarrollo de la presente tesis se ha escogido la metodología HEFESTO, puesto que es la que está orientada principalmente a la construcción de un Data Warehouse que es el principal objetivo de este capítulo.

## 5.4 Aplicación de la metodología HEFESTO

A continuación se detallan cada uno de los pasos de HEFESTO para la construcción del Data Warehouse para los resultados obtenidos en el MOOC.

### 5.4.1 Análisis de requerimientos

A continuación, se describen los distintos parámetros que permitirán conformar el modelo conceptual de datos inicial:

#### 5.4.1.1 Identificar preguntas

En base a los resultados obtenidos con la puesta en marcha del MOOC y las evaluaciones externas (se extenderá una explicación más profunda en el siguiente capítulo), se han identificado las siguientes preguntas para la construcción del Data Warehouse:

- **Pregunta 1:** Número de alumnos clasificados según su edad y ciudad a la que pertenecen.
- **Pregunta 2:** Número de alumnos clasificados por la cantidad de actividades realizadas en el MOOC.
- **Pregunta 3:** Número de alumnos clasificados por el resultado de la participación de los alumnos en el MOOC según su estilo de aprendizaje.
- **Pregunta 4:** Promedio de calificaciones obtenidas en el MOOC según los estilos de aprendizaje de los alumnos.
- **Pregunta 5:** Promedio de calificaciones obtenidas en la evaluación según los estilos de aprendizaje de los alumnos.



- **Pregunta 6:** Número de alumnos correspondientes a cada estilo de aprendizaje según el estado, sexo o docente al que pertenecen.
- **Pregunta 7:** Número de alumnos clasificados por el resultado obtenido en el MOOC según estilos de aprendizaje, conocimientos previos adquiridos, estado o docente al que pertenecen.
- **Pregunta 8:** Promedio de calificaciones obtenidas en la parte teórica de la evaluación según el estado y tipo de alumno.
- **Pregunta 9:** Promedio de calificaciones obtenidas en la parte práctica de la evaluación según el estado y tipo de alumno.

#### 5.4.1.2 Identificar indicadores y perspectivas

##### Pregunta 1

-Número de alumnos clasificados según su edad y ciudad a la que pertenecen.

INDICADOR

PERSPECTIVA

##### Pregunta 2

-Número de alumnos clasificados por la cantidad de actividades realizadas en el MOOC.

INDICADOR

PERSPECTIVA

##### Pregunta 3

-Número de alumnos clasificados por el resultado de la participación en el MOOC según su

INDICADOR

PERSPECTIVA

estilo de aprendizaje.

PERSPECTIVA

##### Pregunta 4

-Promedio de calificaciones obtenidas en el MOOC según los estilos de aprendizaje de los

INDICADOR

PERSPECTIVA

alumnos.

##### Pregunta 5

- Promedio de calificaciones obtenidas en la evaluación según los estilos de aprendizaje de

INDICADOR

PERSPECTIVA

los alumnos.

**Pregunta 6**

-Número de alumnos correspondientes a cada estilo de aprendizaje según el estado, sexo o docente al que pertenecen.

**INDICADOR** **PERSPECTIVAS**

**PERSPECTIVA**

**Pregunta 7**

-Número de alumnos clasificados por el resultado obtenido en la evaluación (teoría, práctica, global) según estilos de aprendizaje, conocimientos previos adquiridos, estado o docente al que pertenecen.

**INDICADOR** **PERSPECTIVAS**

**PERSPECTIVAS**

**Pregunta 8**

-Promedio de calificaciones obtenidas en la parte teórica de la evaluación según el estado y tipo de alumno.

**INDICADOR** **PERSPECTIVA**

**PERSPECTIVA**

**Pregunta 9**

-Promedio de calificaciones obtenidas en la parte práctica de la evaluación según el estado y tipo de alumno.

**INDICADOR** **PERSPECTIVA**

**PERSPECTIVA**

A continuación se detallan los indicadores encontrados:

- Número de alumnos
- Promedio de calificaciones en el MOOC
- Promedio de calificaciones en la evaluación teórica
- Promedio de calificaciones en la evaluación práctica
- Promedio de calificación global de la evaluación

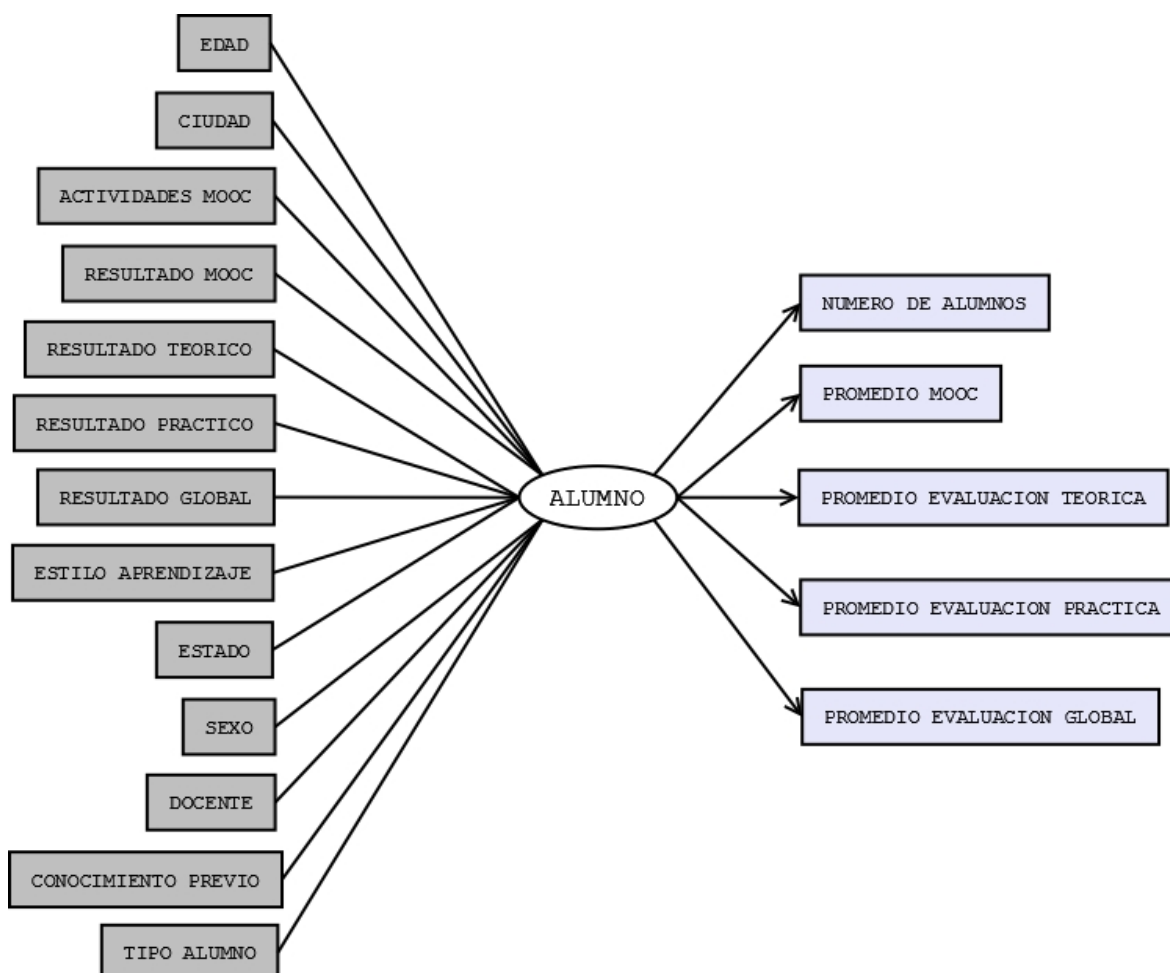
Mientras que entre las perspectivas se tiene:

- Edad
- Ciudad

- Actividades en el MOOC
- Resultado MOOC
- Resultado evaluación teórica
- Resultado evaluación práctica
- Resultado evaluación global
- Estilo de aprendizaje
- Estado
- Sexo
- Docente
- Conocimientos previos
- Tipo de alumno

#### 5.4.1.3 Modelo Conceptual

Una vez obtenido los indicadores y las perspectivas de análisis se procede a crear el modelo conceptual. La *Figura 5.6* muestra el modelo obtenido.



*Figura 5.6: Modelo Conceptual*  
Fuente: Elaboración propia

#### 5.4.2 Análisis de fuentes de datos

A continuación, se establece una relación entre las fuentes de datos disponibles y el modelo conceptual creado en el paso anterior:



### 5.4.2.1 Conformar Indicadores

#### Número de alumnos

- Hechos: NotaMOOC
- Función de conteo: Count
- Aclaración: este indicador representará el conteo de alumnos que cumplen con los parámetros de análisis requeridos.

#### Promedio MOOC

- Hechos: NotaMOOC
- Función de conteo: Average
- Aclaración: este indicador representará el promedio de las calificaciones obtenidas por los alumnos en el MOOC.

#### Promedio evaluación teórica

- Hechos: EvaluacionTeorico
- Función de conteo: Average
- Aclaración: este indicador representará el promedio de las calificaciones obtenidas por los alumnos en la evaluación teórica.

#### Promedio evaluación practica

- Hechos: EvaluacionPractico
- Función de conteo: Average
- Aclaración: este indicador representará el promedio de las calificaciones obtenidas por los alumnos en la evaluación práctica.

#### Promedio evaluación global

- Hechos: EvaluacionGlobal
- Función de conteo: Average
- Aclaración: este indicador representará el promedio de las calificaciones obtenidas por los alumnos en la evaluación práctica sumado a la evaluación teórica.

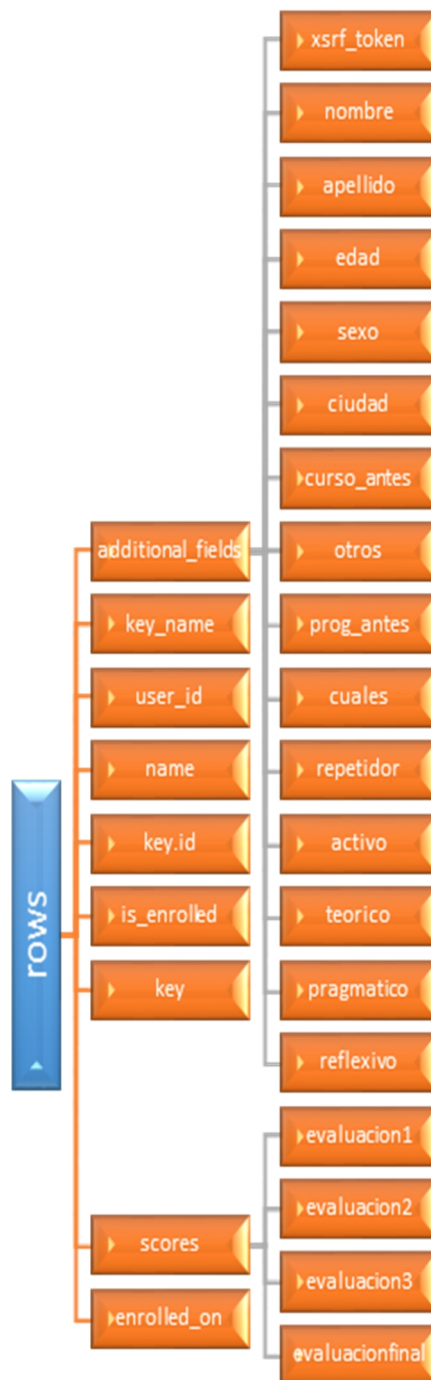
### 5.4.2.2 Establecer correspondencias

Para establecer las correspondencias respectivas del modelo conceptual con las fuentes de datos, a continuación se describen las fuentes de datos que se han considerado para la construcción del Data Warehouse.

- **Información de los estudiantes que tomaron el MOOC:** Esta información se encuentra almacenada en un archivo JSON obtenido del MOOC. En el cual consta toda la información del registro de los estudiantes así como las notas de las evaluaciones obtenidas en el MOOC. La estructura de este archivo se puede observar en la *Figura 5.7*.
- **Encuesta aplicada a los estudiantes:** Esta información se encuentra almacenada en un archivo de Excel y corresponde a una encuesta aplicada a los estudiantes que no cursaron el MOOC realizada por medio de Google Forms. La estructura de este archivo se la puede visualizar en la *Figura 5.8*.

- **Notas de los estudiantes:** Esta fuente de datos corresponde a un archivo de Excel que contiene las notas obtenidas por los estudiantes en la evaluación final que se tomó tanto a los estudiantes que cursaron el MOOC como a los estudiantes que asistieron a las clases tradicionales. La estructura de este archivo se lo puede observar en la *Figura 5.9*.

Una vez que se han definido las fuentes de datos que se van a utilizar para la creación del Data Warehouse, se procede a establecer la correspondencia respectiva con el modelo conceptual.



*Figura 5.7: Estructura del archivo JSON obtenido del MOOC.  
Fuente: Elaboración propia*



ARCHIVO DE ENCUESTA
<ul style="list-style-type: none"> <li>• Marca temporal</li> <li>• Nombres</li> <li>• Apellidos</li> <li>• Edad</li> <li>• Sexo</li> <li>• Ciudad de origen</li> <li>• ¿Ha tomado antes un curso o materia de programación?</li> <li>• En caso de que su respuesta sea afirmativa, ¿Dónde?</li> <li>• ¿Programa en algún lenguaje de programación?</li> <li>• En caso de que su respuesta sea afirmativa, ¿Cuál?</li> <li>• ¿Es repetidor de la asignatura?</li> <li>• Activo</li> <li>• Reflexivo</li> <li>• Pragmático</li> <li>• Teórico</li> <li>• ¿Cuál es su profesor de la asignatura de Programación?</li> <li>• PRACTICA</li> <li>• TEORIA</li> <li>• ExamenEscrito</li> </ul>

Figura 5.8: Estructura de la encuesta aplicada a los estudiantes que no cursaron el MOOC.

Fuente: Elaboración propia.

ARCHIVO CALIFICACIONES
<ul style="list-style-type: none"> <li>• NOMBRE</li> <li>• APELLIDO</li> <li>• DOCENTE</li> <li>• TEORICO</li> <li>• PRACTICO</li> <li>• EXAMENESCRITO</li> </ul>

Figura 5.9: Estructura de las notas de los estudiantes.

Fuente: Elaboración propia.

En la *Figura 5.10* se puede observar las correspondencias entre los indicadores y las fuentes de datos, mientras que en la *Figura 5.11* se puede observar las correspondencias entre las perspectivas y las fuentes de datos.

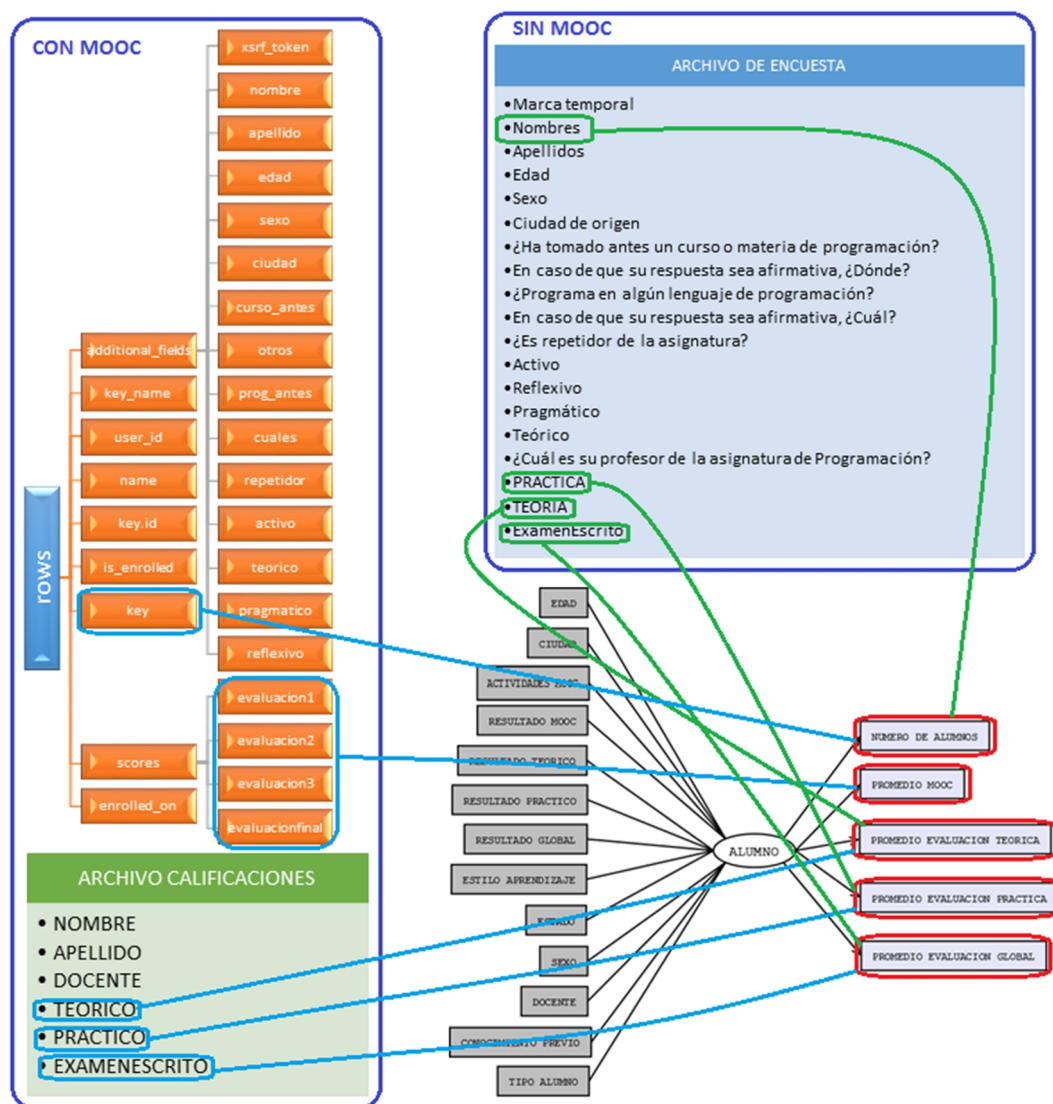


Figura 5.10: Correspondencia de los indicadores y las fuentes de datos.  
Fuente: Elaboración propia

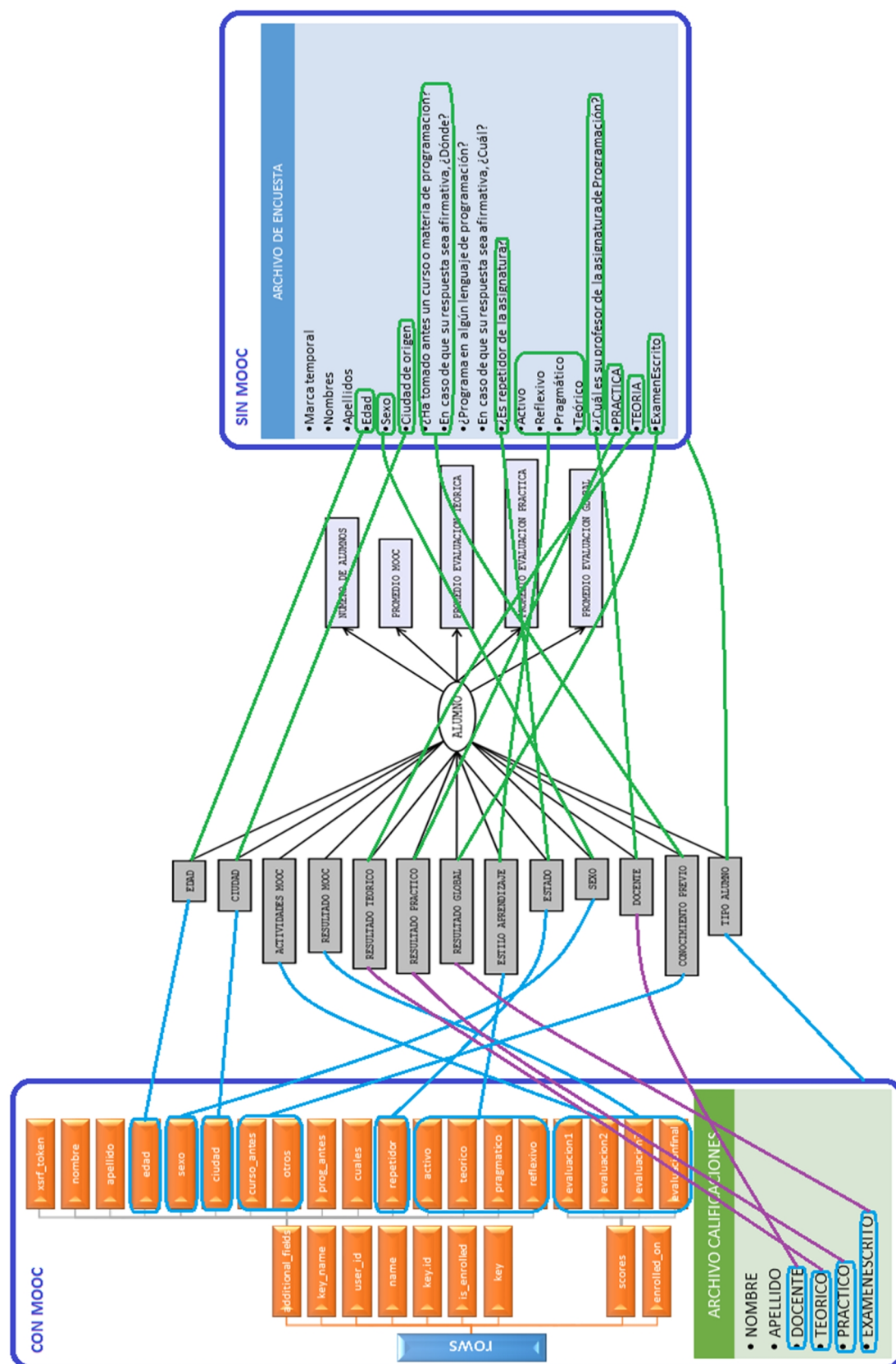


Figura 5.11: Correspondencia de perspectivas y las fuentes de datos.  
Fuente: Elaboración propia



### 5.4.2.3 Modelo Conceptual Ampliado

**Perspectiva Edad:** representa la edad que posee el alumno.

- **DescripcionEdad:** representa el número de años que posee el alumno.

**Perspectiva Ciudad:** representa la ciudad a la que pertenece el alumno

- **NombreCiudad:** nombre de la ciudad

**Perspectiva Actividades en el MOOC:** representa la cantidad de actividades realizadas por el estudiante dentro del MOOC.

- **Cantidad:** número de actividades.

**Perspectiva Resultado MOOC:** representa el resultado del alumno obtenido en el MOOC.

- **DescripcionRM:** cuyos valores serán Aprobado (60-100), Reprobado (0-59) o Desertado (no rindió el examen final del MOOC) en caso de que el alumno haya abandonado el MOOC.

**Perspectiva Resultado evaluación teórica:** es una descripción cualitativa de la calificación obtenida por el alumno en la evaluación teórica.

- **DescripcionRET:** cuyos valores serán Insuficiente (0-4), Regular (5-6), Buena (7-8), Muy Buena (9) y Excelente (10).

**Perspectiva Resultado evaluación práctica:** es una descripción cualitativa de la calificación obtenida por el alumno en la evaluación práctica.

- **DescripcionREP:** cuyos valores serán Insuficiente (0-4), Regular (5-6), Buena (7-8), Muy Buena (9) y Excelente (10).

**Perspectiva Resultado evaluación global:** es una descripción cualitativa de la calificación obtenida por el alumno en la evaluación global, es decir, la suma de las calificaciones obtenidas en la evaluación teórica y en la evaluación práctica.

- **DescripcionREG:** cuyos valores serán Insuficiente (0-8), Regular (9-12), Buena (13-15), Muy Buena (16-18) y Excelente (19-20).

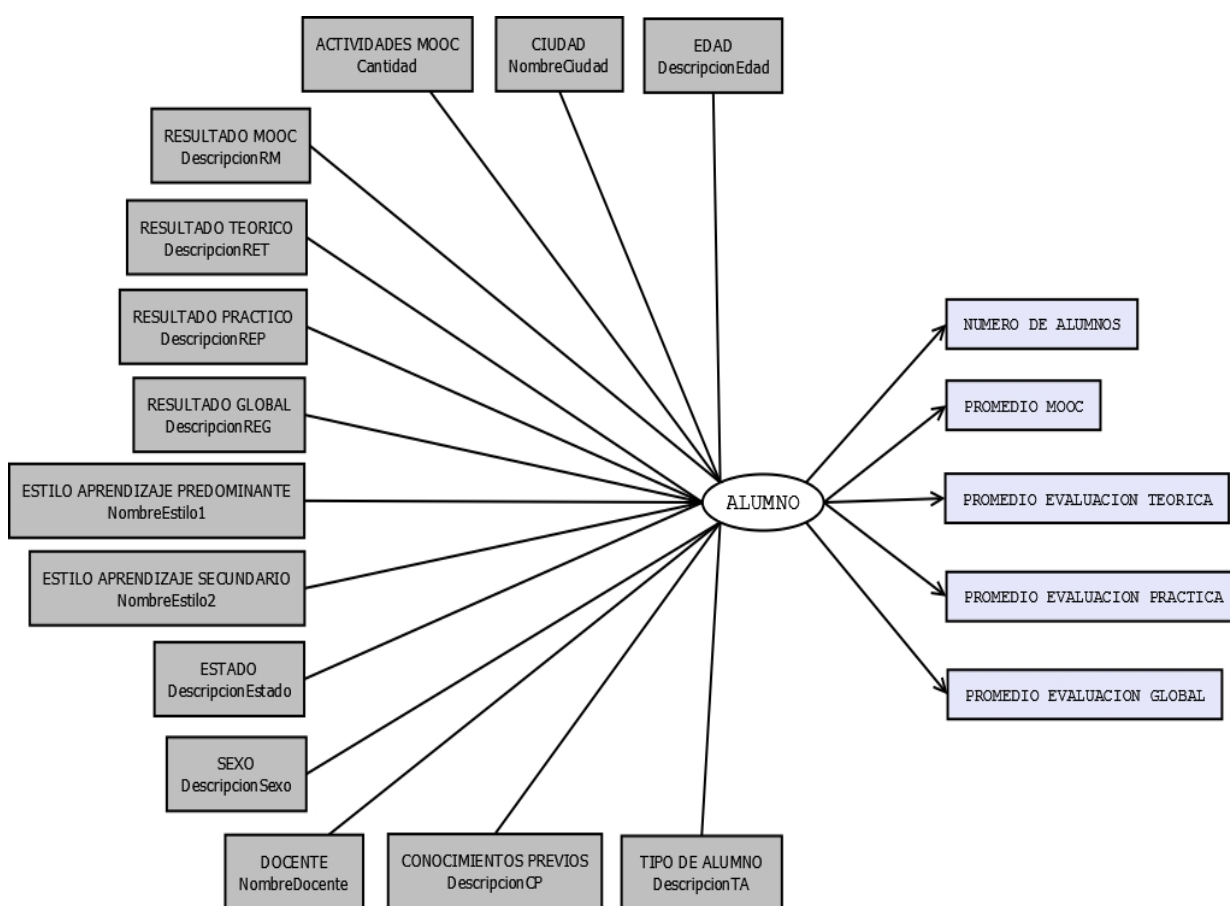
#### **Perspectiva Estilo de aprendizaje**

Para poder determinar el estilo de aprendizaje predominante del alumno se compara los valores obtenidos por los alumnos en cada estilo y se toma el más alto, puesto que hay alumnos en los que existen iguales valores para diferentes estilos de aprendizaje se ha decidido tomar los dos estilos de aprendizaje de mayor valor y crear dos perspectivas denominadas de la siguiente manera:

- **Perspectiva Estilo de aprendizaje predominante:**
  - **NombreEstilo1:** cuyo posible valor es Activo, Reflexivo, Pragmático y Teórico
- **Perspectiva Estilo de aprendizaje secundario:**
  - **NombreEstilo2:** cuyo posible valor es Activo, Reflexivo, Pragmático y Teórico
- **Perspectiva Estado:** indica la estancia actual del alumno en la asignatura.

- DescripcionEstado: cuyo posible valor es Repetidor o Nuevo.
- **Perspectiva Sexo: representa el sexo del alumno.**
  - DescripcionSexo: cuyo posible valor es Masculino o Femenino.
- **Perspectiva Docente: indica el docente que está a cargo del alumno en la asignatura.**
  - NombreDocente: es el respectivo nombre del docente.
- **Perspectiva Conocimientos Previos: indica si el alumno ha tenido conocimientos previos de la asignatura.**
  - DescripcionCP: cuyo posible valor puede ser Colegio, Instituto o No.
- **Perspectiva Tipo de alumno**
  - DescripcionTA: cuyo posible valor puede ser CON MOOC, SIN MOOC.

Después de realizar el respectivo análisis se puede observar en la *Figura 5.12* el Modelo Conceptual Expandido:



*Figura 5.12: Modelo conceptual expandido.  
Fuente: Elaboración propia*

### 5.4.3 Modelo lógico del Data Warehouse

A continuación, se convierte los parámetros del modelo conceptual a tablas de bases de datos:

### 5.4.3.1 Tipo del modelo lógico del Data Warehouse

El tipo de modelo lógico seleccionado para seguir adelante con el proceso de creación del Data Warehouse es el de estrella, este tipo es compatible con las herramientas seleccionadas para la implementación del Data Warehouse.

#### TABLAS DE DIMENSIONES

##### Perspectiva Edad

- La nueva tabla de dimensión tendrá el nombre "EDAD".
- Se le agregará una clave principal con el nombre "idEdad".
- El nombre del campo "DescripcionEdad" será cambiado por "descripcionEdad".

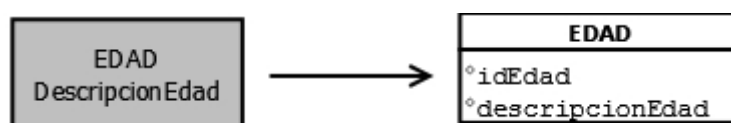


Figura 5.13: Tabla Dimensión Edad.  
Fuente: Elaboración propia

##### Perspectiva Ciudad

- La nueva tabla de dimensión tendrá el nombre "CIUDAD".
- Se le agregará una clave principal con el nombre "idCiudad".
- El nombre del campo "NombreCiudad" será cambiado por "nombreCIUDAD".

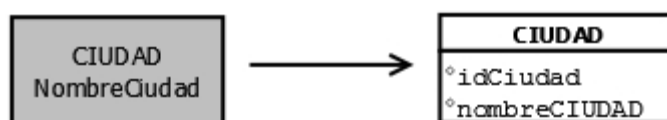


Figura 5.14: Tabla Dimensión Ciudad.  
Fuente: Elaboración propia

##### Perspectiva Actividades en el MOOC

- La nueva tabla de dimensión tendrá el nombre "ACTIVIDADES".
- Se le agregará una clave principal con el nombre "idActividades".
- El nombre del campo "Cantidad" será cambiado por "cantidad".

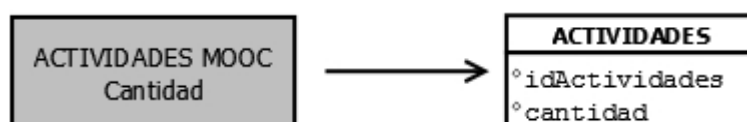


Figura 5.15: Tabla Dimensión Actividades.  
Fuente: Elaboración propia

##### Perspectiva Resultado MOOC

- La nueva tabla de dimensión tendrá el nombre "RESULTADOMOOC".
- Se le agregará una clave principal con el nombre "idResultadoMOOC".
- El nombre del campo "DescripcionRM" será cambiado por "descripcionRM".

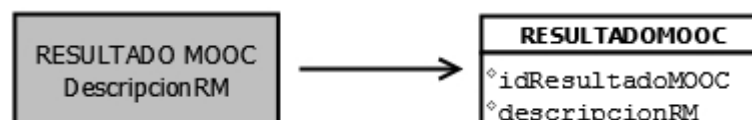


Figura 5.16: Tabla Dimensión ResultadoMOOC.

Fuente: Elaboración propia

**Perspectiva Resultado evaluación teórica**

- La nueva tabla de dimensión tendrá el nombre "EVALUACIONTEORICA".
- Se le agregará una clave principal con el nombre "idEvaluacionTeorica".
- El nombre del campo "DescripcionRET" será cambiado por "descripcionRET".

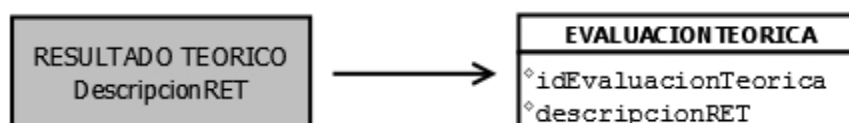


Figura 5.17: Tabla Dimensión EvaluacionTeorica.

Fuente: Elaboración propia

**Perspectiva Resultado evaluación práctica**

- La nueva tabla de dimensión tendrá el nombre "EVALUACIONPRACTICA".
- Se le agregará una clave principal con el nombre "idEvaluacionPractica".
- El nombre del campo "DescripcionREP" será cambiado por "descripcionREP".

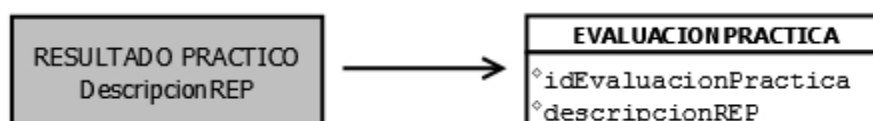


Figura 5.18: Tabla Dimensión EvaluacionPractica.

Fuente: Elaboración propia

**Perspectiva Resultado evaluación global**

- La nueva tabla de dimensión tendrá el nombre "RESULTADOEVALUACION".
- Se le agregará una clave principal con el nombre "idResultadoEvaluacion".
- El nombre del campo "DescripcionREG" será cambiado por "descripcionREG".

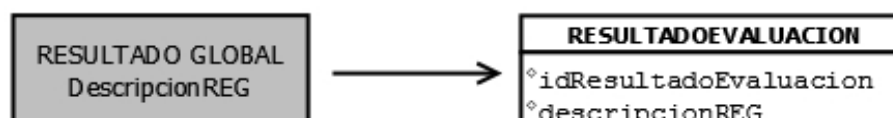


Figura 5.19: Tabla Dimensión ResultadoEvaluacion.

Fuente: Elaboración propia

**Perspectiva Estilo de aprendizaje predominante**

- La nueva tabla de dimensión tendrá el nombre "ESTILOPREDOMINANTE".
- Se le agregará una clave principal con el nombre "idEstiloPredominante".
- El nombre del campo "NombreEstilo1" será cambiado por "nombreEstilo1".





Figura 5.20: Tabla Dimensión EstiloPredominante.

Fuente: Elaboración propia

**Perspectiva Estilo de aprendizaje secundario**

- La nueva tabla de dimensión tendrá el nombre “ESTILOSECUNDARIO”.
- Se le agregará una clave principal con el nombre “idEstiloSecundario”.
- El nombre del campo “NombreEstilo2” será cambiado por “nombreEstilo2”.

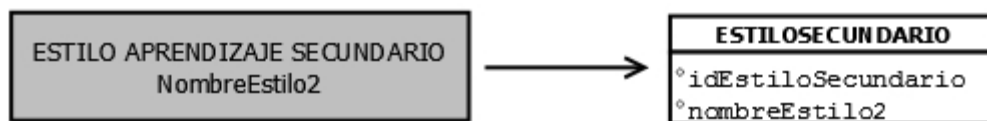


Figura 5.21: Tabla Dimensión EstiloSecundario.

Fuente: Elaboración propia

**Perspectiva Estado**

- La nueva tabla de dimensión tendrá el nombre “ESTADO”.
- Se le agregará una clave principal con el nombre “idEstado”.
- El nombre del campo “DescripcionEstado” será cambiado por “descripcionEstado”.

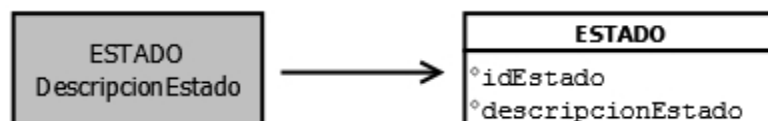


Figura 5.22: Tabla Dimensión Estado.

Fuente: Elaboración propia

**Perspectiva Sexo**

- La nueva tabla de dimensión tendrá el nombre “SEXO”.
- Se le agregará una clave principal con el nombre “idSexo”.
- El nombre del campo “DescripcionSexo” será cambiado por “descripcionSexo”.

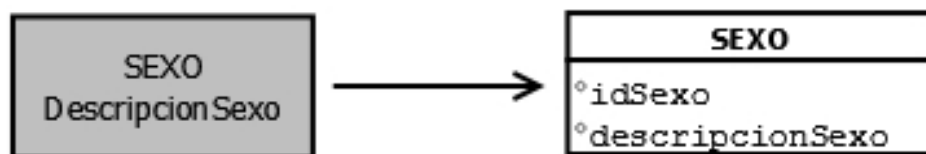


Figura 5.23: Tabla Dimensión Sexo.

Fuente: Elaboración propia

**Perspectiva Docente**

- La nueva tabla de dimensión tendrá el nombre “DOCENTE”.
- Se le agregará una clave principal con el nombre “idDocente”.
- El nombre del campo “NombreDocente” será cambiado por “nombreDocente”.

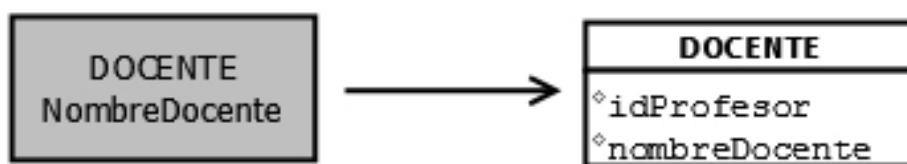


Figura 5.24: Tabla Dimensión Docente.  
Fuente: Elaboración propia

### Perspectiva Conocimientos Previos

- La nueva tabla de dimensión tendrá el nombre “CONOCIMIENTOPREVIO”.
- Se le agregará una clave principal con el nombre “idConocimientoPrevio”.
- El nombre del campo “DescripcionCP” será cambiado por “descripcionCP”.

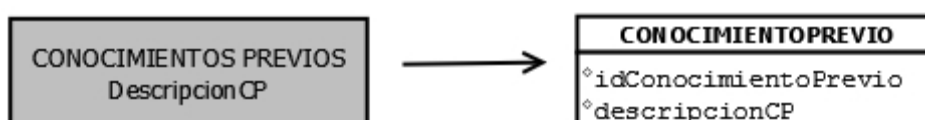


Figura 5.25: Tabla Dimensión ConocimientoPrevio.  
Fuente: Elaboración propia

### Perspectiva Tipo de alumno

- La nueva tabla de dimensión tendrá el nombre “TIPOALUMNO”.
- Se le agregará una clave principal con el nombre “idTipoAlumno”.
- El nombre del campo “DescripcionTA” será cambiado por “descripcionTA”.

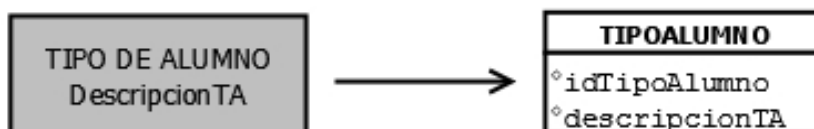


Figura 5.26: Tabla Dimensión TipoAlumno.  
Fuente: Elaboración propia

## TABLAS DE HECHOS

- La tabla de hechos tendrá el nombre “ALUMNO”
- Su clave principal será la combinación de las claves principales de las tablas de dimensiones antes definidas: “idEdad”, “idCiudad”, “idActividades”, “idResultadoMOOC”, “idEvaluacionTeorica”, “idEvaluacionPractica”, “idResultadoEvaluacion”, “idEstiloPredominante”, “idEstiloSecundario”, “idEstado”, “idSexo”, “idDocente”, “idConocimientoPrevio”, “idTipoAlumno”.
- Se crearán cuatro hechos, que corresponden a los indicadores creados anteriormente: “NotaMOOC”, “EvaluacionPractico”, “EvaluacionTeorico” y “EvaluacionGlobal”.

La Figura 5.27 muestra la tabla de hechos creada.

### 5.4.3.2 Uniones

La Figura 5.28 muestra el modelo obtenido a través de las uniones de la tabla de hechos con las tablas de dimensiones.

ALUMNO
◦idEdad
◦idSexo
◦idCiudad
◦idTipoAlumno
◦idDocente
◦idActividades
◦idResultadoMOOC
◦idEvaluacionTeorica
◦idEvaluacionPractica
◦idResultadoEvaluacion
◦idEstiloPredominante
◦idEstiloSecundario
◦idEstado
◦idConocimientoPrevio
◦NotaMOOC
◦EvaluacionPractico
◦EvaluacionTeorico
◦EvaluacionGlobal

Figura 5.27: Tabla de hechos Alumno.  
Fuente: Elaboración propia

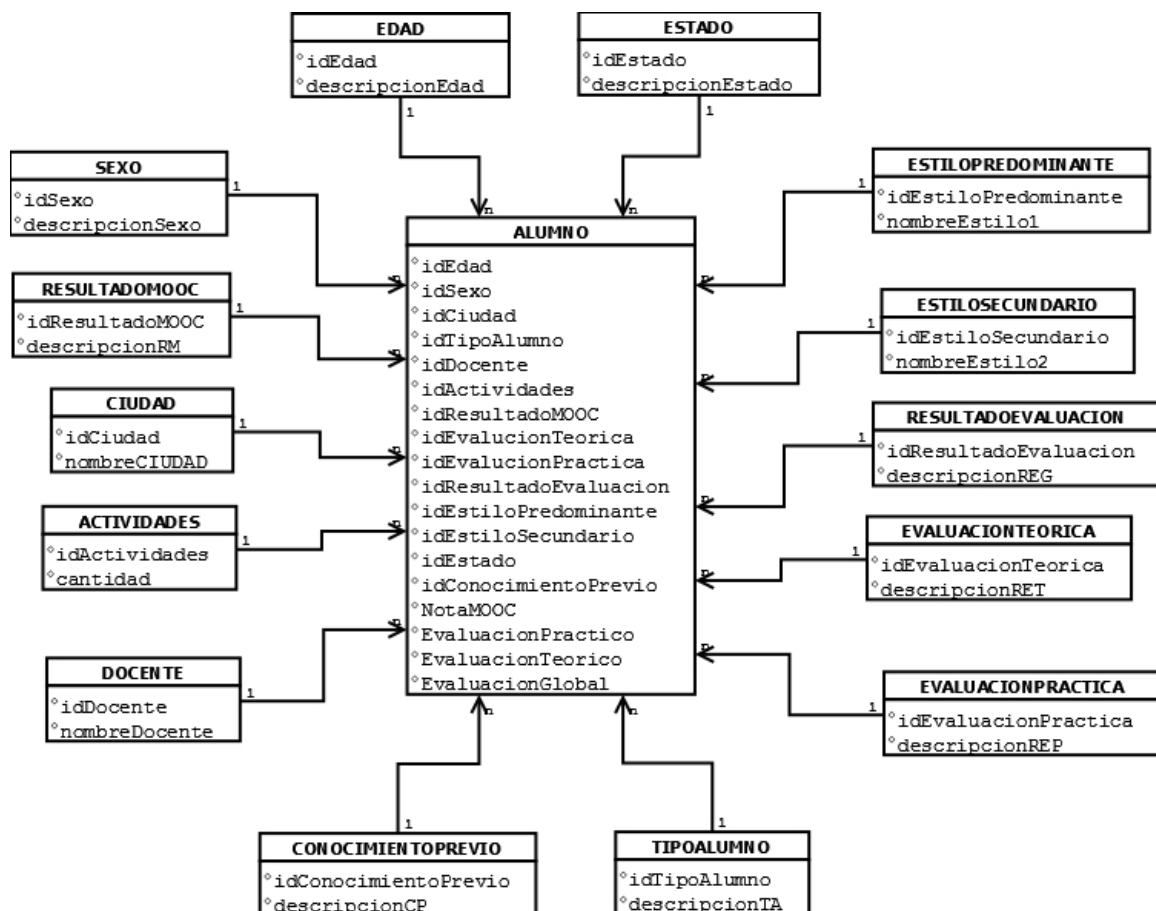


Figura 5.28: Modelo lógico de Alumno.  
Fuente: elaboración propia



## 5.4.4 Integración de datos

A continuación, como parte final se carga el Data Warehouse con datos:

### 5.4.4.1 Carga Inicial

#### Proceso ETL

Una vez creado el modelo conceptual de datos se procede a realizar la transformación que permitirá llevar los datos de las distintas fuentes al Data Warehouse, el cual estará alojado en un servidor MYSQL. Como existen tres fuentes de datos distintas es necesario tratar los datos (buscar errores, seleccionar campos, filtrar información) antes de poder unirlos en un solo flujo. La transformación se lleva a cabo a través de la herramienta Pentaho Data-Integration.

Como primer paso se toman los datos que se han extraído desde Google Course Builder y que se encuentran en un archivo JSON, estos datos una vez que son tratados se unen a los datos obtenidos de un archivo externo; el archivo externo contiene las notas de las evaluaciones de los alumnos. Finalmente, se toman los datos arrojados por la encuesta realizada a los alumnos que no tomaron el MOOC, estos datos son procesados y enviados para formar parte de un solo flujo con los datos de los alumnos que si tomaron el MOOC, estos datos resultantes poblarán el Data Warehouse. La *Figura 5.29* muestra la transformación completa en Pentaho Data-Integration.

#### Creación del cubo multidimensional

Pentaho BI-Server crea un cubo multidimensional por defecto a partir de la conexión y parametrización de la base de datos que contiene el Data Warehouse. Pentaho BI-Server permite modificar el cubo multidimensional de tal forma que se pueda agregar o quitar los indicadores y las dimensiones. La *Figura 5.30* muestra la edición de la estructura del cubo multidimensional.

El cubo multidimensional nos permitirá generar los reportes para obtener los resultados en el caso de estudio que se realizará en el siguiente capítulo.

## 5.5 Recapitulación

En el presente capítulo se realizó una revisión acerca de los Data Warehouse y sus metodologías de desarrollo, en donde, se escogió la metodología de Hefesto por considerarse sencilla y adaptable a las necesidades de estudio de este trabajo de tesis. Una vez que se creó el modelo de datos conceptual se empleó una transformación mediante Pentaho Data-Integration la cual permitió limpiar y procesar los datos para luego poblar el Data Warehouse. Finalmente, a través de Pentaho BI-Server se editó el cubo que se había creado por defecto, lo cual permite realizar los correspondientes análisis presentes en el próximo capítulo.



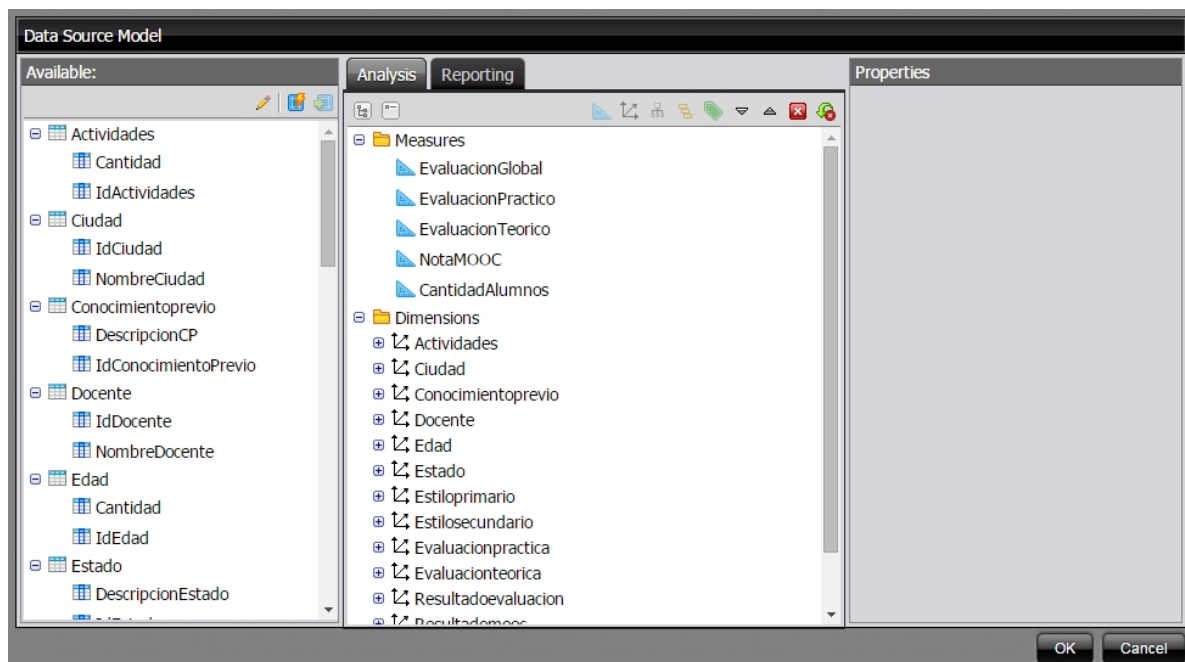


Figura 5.30: Estructura del cubo multidimensional del BI-Server.  
Fuente: Elaboración propia



# CAPÍTULO 6:

## Caso de estudio

### 6.1 Introducción

Después de haber creado y puesto en marcha el MOOC es importante recoger y analizar la información generada tras la finalización del mismo. Por esta razón se ha visto conveniente la creación de un caso de estudio basado en la información obtenida del MOOC, en dónde como medio de análisis de información se utiliza el Data Warehouse creado en el capítulo previo. También se abordan temas acerca de la aplicación de la minería de datos educacional, en donde se busca encontrar la utilidad de la misma sobre la información manejada por el caso de estudio, de tal manera que el modelo conceptual generado en el capítulo previo pueda servir de referencia para futuros estudios de la aplicación de los MOOCs en la educación.

### 6.2 Selección y definición del caso de estudio

Se ha decidido emplear el MOOC creado como producto de este trabajo de tesis para poder conocer su eficiencia dentro de la enseñanza y por otro lado también entender cómo influye un MOOC y la presentación de sus contenidos en el aprendizaje de los estudiantes, tomando en cuenta los diferentes estilos de aprendizaje que se pueden presentar entre estos estudiantes. Entre los distintos estilos de aprendizaje tenemos: activo, pragmático, teórico y reflexivo.

#### 6.2.1 Ámbitos en los que es relevante el estudio

El estudio del presente caso permitirá mostrar el verdadero valor que puede tener un MOOC en la enseñanza, dado que si los resultados arrojados por este estudio son positivos los MOOCs podrían ser empleados por la Universidad de Cuenca como un medio para poder difundir conocimiento, no solo a estudiantes de la institución sino también a sujetos externos a ella, por otra parte también debería considerarse a los MOOCs como una herramienta auxiliar para el aprendizaje de los alumnos.



## 6.2.2 Justificación

Cada semestre la Facultad de Ingeniería de la Universidad de Cuenca oferta la asignatura de Algoritmos, Datos y Estructuras como parte de las asignaturas de tipo Ciencias Básicas, estas asignaturas son comunes en todas las carreras de la Facultad de Ingeniería. El ofertar la asignatura de Algoritmos, Datos y Estructuras en todas las carreras de la facultad conlleva a tener un número amplio de alumnos, este número de alumnos está distribuido en una cantidad de 5 cursos con un promedio de 30 alumnos por curso. Atender las necesidades cognitivas de esta cantidad de alumnos se puede volver tedioso para los docentes encargados de los distintos cursos, teniendo como consecuencia la obtención de bajas calificaciones por parte de los alumnos, pudiéndoles llevar a los alumnos a la suspensión de la materia o en casos más graves a la reprobación. Es por esta razón que se debe buscar nuevas formas que permitan mejorar el proceso de transmisión de la información. Entre estas formas se pretende introducir los MOOCs dentro de la enseñanza presencial y evidenciar medir su eficacia.

## 6.2.3 Objetivo del caso de estudio

El objetivo del caso de estudio es reunir evidencia sobre el uso de MOOC como apoyo a las clases presenciales y cómo impacta esto sobre el rendimiento académico de los estudiantes, utilizando un modelo conceptual que integre distintos orígenes de datos y que permita interpretar la información obtenida

# 6.3 Fundamentación teórica

En esta sección se presentan conceptos que se manejan en este capítulo.

## 6.3.1 Estilos de Aprendizaje

Uno de los principales parámetros que se han considerado para formar el perfil de los alumnos que formarán parte de este caso de estudio son los estilos de aprendizaje, que hacen referencia a un conjunto de características pedagógicas y cognitivas que suelen expresarse conjuntamente cuando una persona debe enfrentar una situación de aprendizaje (Alonso & Gallego, 2004).

Para ello, se ha tomado como base el modelo de estilos de aprendizaje de Honey y Alonso (Alonso & Gallego, 2004), quienes clasifican a las personas en cuatro estilos de aprendizaje:

- **Activos:** son personas que se implican plenamente y sin prejuicios en nuevas experiencias. Son de mente abierta, nada escépticos y acometen con entusiasmo en tareas nuevas. El trabajo es de grupo y se involucran en los asuntos de los demás y centran a su alrededor todas las actividades.
- **Reflexivos:** son personas a las que les gusta considerar las experiencias y observarlas desde diferentes perspectivas. Recogen datos, analizándolos con detenimiento antes de llegar a alguna conclusión. Les gusta considerar todas las alternativas posibles antes de realizar una acción.
- **Teóricos:** los teóricos adaptan e integran las observaciones dentro de teorías lógicas y complejas. Tienden a ser perfeccionistas. Integran los hechos en teorías coherentes. Les gusta analizar y sintetizar. Son profundos en su sistema de pensamiento, a la hora de establecer principios, teorías y modelos.
- **Pragmáticos:** el punto fuerte de las personas con predominancia de este estilo de aprendizaje es la aplicación práctica de las ideas. Descubren el aspecto positivo de las nuevas ideas y aprovechan la primera oportunidad para experimentarlas. Actúan



rápidamente y con seguridad con aquellas ideas y proyectos que les atrae (Alonso & Gallego, 2004).

Para determinar los estilos de aprendizaje de cada uno de los estudiantes se les solicitó que realicen el *cuestionario de Honey-Alonso de Estilos de Aprendizaje*, disponible en <http://www.estilosdeaprendizaje.es/chaee/chaee.htm>.

### 6.3.2 Minería de datos en la educación

Existen diversas definiciones de data mining, una de ellas las considera como “extracción no trivial de información implícita, previamente no conocida y potencialmente útil desde los datos” (Hipp, Guntzer, & Nakhaeizadeh, 2000).

Si se considera una definición más técnica, se considera como un proceso analítico diseñado para explorar grandes cantidades de datos en búsqueda de patrones consistentes y/o relaciones sistemáticas entre variables, para luego validar el hallazgo aplicando los patrones encontrados a nuevos conjuntos de datos (Corso & Alfaro, 2014).

La tecnología de data mining permite generar nuevas oportunidades de negocios, ofreciendo las siguientes características (Corso & Alfaro, 2014):

- Predicción de tendencias y comportamientos: Estos modelos toman como base un hipótesis definida por el usuario para posteriormente efectuar una prueba de validación contra los datos, con el fin de evaluar la calidad de estos modelos, tales como la simplicidad, interoperabilidad, etc.
- Modelos de descubrimiento: Este tipo de modelos intentan encontrar relaciones y patrones de comportamiento en el conjunto de datos para ofrecer conocimiento sobre un problema concreto.

La minería de datos<sup>23</sup> en el campo de la educación (Educational Datamining, EDM) se ha convertido en una herramienta muy útil especialmente cuando se trata de examinar el comportamiento del aprendizaje de los estudiantes en entornos de aprendizaje en línea. Esto se debe al potencial de la minería de datos en el análisis y descubrimiento de la información oculta de los datos que se volvería un procedimiento difícil y muy lento de realizar manualmente (Khadijah Mohamad & Zaidatun, 2013).

EDM es una disciplina emergente que nace de la necesidad de desarrollar métodos para la exploración de datos provenientes de centros educativos, y el uso de los mismos para entender mejor a los estudiantes y generar información que permita mejorar la forma en la que se enseña. Algunas áreas claves de EDM son:

- Desempeño del estudiante
- Inscripciones de los estudiantes.
- Predicción del rendimiento de los estudiantes
- Estudio del tipo de aprendizaje con el fin de recomendar mejoras en la práctica educativa.

---

<sup>23</sup> Minería de datos, conocido también como Datamining, es el proceso de detectar la información procesable de los conjuntos grandes de datos. Utiliza el análisis matemático para deducir los patrones y tendencias que existen en los datos. Normalmente, estos patrones no se pueden detectar mediante la exploración tradicional de los datos porque las relaciones son demasiado complejas o porque hay demasiados datos (MSDN, 2014).



EMD es considerada una ciencia del aprendizaje así como también un área de la minería de datos (Goyal & Vohra, 2012).

Las principales aplicaciones de EDM se listan a continuación:

- **Análisis y visualización de datos:** Se utiliza para resaltar información útil para la toma de decisiones. En el ámbito educativo, por ejemplo, ayuda a los educadores y administradores del curso a analizar las actividades realizadas por los estudiantes en el curso y maneja la información para obtener una visión general de aprendizaje de los estudiantes.
- **Predecir el rendimiento de los estudiantes:** En este caso, se estima el valor desconocido de una variable que describe al estudiante. En la educación, los valores normalmente previstos son el desempeño de los estudiantes, sus conocimientos, calificaciones, etc. Este valor puede ser numérico/continuo o categórico/discreto.
- **Análisis de valores atípicos:** La detección de valores atípicos se utiliza para detectar y eliminar observaciones anómalas de datos; además puede identificar las fallas del sistema y el fraude antes de que produzcan consecuencias potencialmente adversas.
- **Agrupación de estudiantes:** Permite agrupar a los estudiantes de acuerdo a sus características personalizadas, personales, etc. Estos grupos de estudiantes pueden ser utilizadas por el docente para construir un sistema de aprendizaje personalizado que puede promover el aprendizaje de grupo efectivo. Las técnicas utilizadas para esta aplicación son la clasificación<sup>24</sup> y la clusterización<sup>25</sup> (Goyal & Vohra, 2012).

## 6.4 Elaboración de preguntas

En esta parte se describen las preguntas que se pretenden responder mediante el caso de estudio:

- ¿Cuáles son los estilos de aprendizaje predominantes dentro de los estudiantes de cada muestra?
- ¿Cuáles son los estilos de aprendizaje según el sexo dentro de los estudiantes de cada muestra?
- ¿En qué muestra se obtienen los mejores resultados?
- ¿Cuáles son los estilos de aprendizaje predominantes en los estudiantes de mayor rendimiento de cada muestra?
- ¿Cuál es el resultado obtenido por alumnos que repiten la materia o tienen conocimientos previos de programación contra los que son completamente nuevos en el tema?
- De los alumnos que repiten la materia dependiendo de si siguieron o no el MOOC ¿quién obtuvo mejor media en lo teórico?
- De los alumnos que repiten la materia dependiendo de si siguieron o no el MOOC ¿quién obtuvo mejor media en lo práctico?
- De los alumnos que no repiten la materia dependiendo de si siguieron o no el MOOC ¿quién obtuvo mejor media en lo teórico?

<sup>24</sup> Algoritmos de clasificación, son aquellos que predicen una o más variables discretas, basándose en otros atributos del conjunto de datos (MSDN, 2014).

<sup>25</sup> Algoritmos de clusterización, conocidos también como algoritmos de segmentación, consiste en dividir los datos en grupos (clústeres) de elementos que tienen propiedades similares (MSDN, 2014).



- De los alumnos que no repiten la materia dependiendo de si siguieron o no el MOOC ¿quién obtuvo mejor media en lo práctico?
- ¿Cuál es el resultado de la participación de los alumnos en el MOOC según los estilos de aprendizaje?
- ¿Cuál es el promedio del total de las calificaciones obtenidas por los alumnos en la evaluación según los estilos de aprendizaje?

## 6.5 Localización de las fuentes de datos

La información necesaria para llevar a cabo el presente estudio será tomada en base a las calificaciones obtenidas por los estudiantes, así como de encuestas que serán llevadas a cabo.

- **Calificaciones**

Serán la recopilación de las notas obtenidas del rendimiento de los alumnos, por un lado las notas obtenidas dentro del MOOC que son fácilmente exportables mediante las herramientas ofrecidas por Google Course Builder; mientras que por el otro lado se tomarán en cuenta las notas obtenidas en una evaluación general que se llevará a cabo con la participación de los estudiantes. Las evaluaciones y la división de los estudiantes en muestras se explican más a detalle en la parte de Diseño y Metodología.

- **Encuestas**

La primera encuesta se realiza a los estudiantes que siguen el MOOC al momento del registro, para observar la estructura y las preguntas de esta encuesta véase la figura 4.4. Mientras que a los estudiantes que no siguen el MOOC se les hace una encuesta externa mediante los Formularios de Google, que permiten crear las encuestas y exportar los resultados para poder analizarlos. Ambas encuestas tienen la finalidad de tomar datos personales de los alumnos, sobre todo los valores de sus estilos de aprendizaje. En la Figura 6.1, se puede observar la estructura y las preguntas de la encuesta realizada a los alumnos que no siguen el MOOC.

## 6.6 Diseño y Metodología

Como fase inicial del caso de estudio se toman las distintas muestras de la población que en este caso son todos los alumnos que toman la asignatura de Algoritmos, Datos y Estructuras, un total de 140 alumnos, pero que no pueden ser tomados aleatoriamente debido a que ya están divididos en cursos; estas serían las muestras tomadas:

- » **M<sub>1</sub>**: 25 alumnos dirigidos por el Docente 1 que siguen el MOOC como medio de transferencia de información y emplean la clase para realizar prácticas.
- » **M<sub>2</sub>**: 34 alumnos dirigidos por el Docente 2 que siguen el MOOC como medio de transferencia de información y emplean la clase para realizar prácticas.
- » **M<sub>3</sub>**: 24 alumnos dirigidos por el Docente 3 que no toman el MOOC y siguen la clase tradicional enfocada en transmitir información.
- » **M<sub>4</sub>**: 31 alumnos dirigidos por el Docente 4 que no toman el MOOC y siguen la clase tradicional enfocada en transmitir información.
- » **M<sub>5</sub>**: 26 alumnos dirigidos por el Docente 5 que no toman el MOOC y siguen la clase tradicional enfocada en transmitir información.



## Cuestionario de Datos

El presente cuestionario pretende reunir información que permita mejorar las metodologías de enseñanza. Se pide completa honestidad.

\*Obligatorio

Nombres: \*

Por Ejemplo: Juan Diego

Apellidos: \*

Por Ejemplo: Pérez Loja

Edad: \*

Por Ejemplo: 18

Sexo: \*

☐ Femenino

☐ Masculino

Ciudad de origen: \*

Por Ejemplo: Cuenca

¿Cuál es su profesor de la asignatura de Programación? \*

¿Ha tomado antes un curso o materia de programación? \*

Fuera del entorno universitario

☐ Si

☐ No

En caso de que su respuesta sea afirmativa, ¿Dónde? \*

☐ Colegio

☐ Institutos

☐ Otros:

¿Programa en algún lenguaje de programación? \*

☐ Si

☐ No

En caso de que su respuesta sea afirmativa, ¿Cuál? \*

Por Ejemplo: Visual Basic

¿Es repetidor de la asignatura? \*

☐ Si

☐ No

## Estilos de Aprendizaje

Vaya al siguiente enlace: <http://www.estilosdeaprendizaje.es/chaee/chaee.htm>

Conteste el cuestionario y anote los resultados para cada uno de los siguientes valores.

Activo: \*

Reflexivo: \*

Pragmático: \*

Teórico: \*

Nunca envíes contraseñas a través de Formularios de Google.

Figura 6.1: Formato de Encuesta a alumnos sin MOOC.

Fuente: Elaboración propia.

Mientras que las diferentes variables que se han definido para este caso de estudio son:

- » **V<sub>1</sub>**: Estilo de aprendizaje predominante=Activo, Reflexivo, Teórico, Pragmático.
- » **V<sub>2</sub>**: Estilo de aprendizaje secundario=Activo, Reflexivo, Teórico, Pragmático.
- » **V<sub>3</sub>**: Sexo= Masculino, Femenino.
- » **V<sub>4</sub>**: Docente= Docente 1, Docente 2, Docente 3, Docente 4, Docente 5.
- » **V<sub>5</sub>**: Ciudad.
- » **V<sub>6</sub>**: Estado= Repetidor, Nuevo.
- » **V<sub>7</sub>**: Conocimientos previos de programación= Si (Colegio, Instituto), No.
- » **V<sub>8</sub>**: Resultado MOOC= Aprobado, Reprobado, Desertado.
- » **V<sub>9</sub>**: Resultado Evaluación Global= Excelente, Muy Buena, Buena, Regular, Insuficiente.
- » **V<sub>10</sub>**: Resultado Evaluación Teórico= Excelente, Muy Buena, Buena, Regular, Insuficiente.
- » **V<sub>11</sub>**: Resultado Evaluación Practico= Excelente, Muy Buena, Buena, Regular, Insuficiente.
- » **V<sub>12</sub>**: Actividades Realizadas= 0, 1, 2, 3.
- » **V<sub>13</sub>**: Edad.
- » **V<sub>14</sub>**: Tipo de Alumno= Con MOOC, Sin MOOC.

La asignatura de Algoritmos, Datos y Estructuras es impartida a los estudiantes de las diferentes muestras, en donde se cubren los tres primeros capítulos de esta asignatura. Con la diferencia que las muestras sin MOOC toman las clases con un enfoque tradicional, es decir, asisten a clases y el profesor transmite la información. Por otro lado, los alumnos de las muestras que siguen el MOOC (generado como parte del Capítulo 4) emplean este como una herramienta de

transferencia de información. Después de haber terminado con los tres capítulos de la asignatura en un tiempo total de dos semanas, a los alumnos de todas las muestras se les evalúa lo aprendido mediante un test. Este test está formado por una parte teórica sobre 10 puntos y otra práctica también sobre 10 puntos. Entonces, con las calificaciones obtenidas en estas evaluaciones y mediante las otras fuentes de información (datos generados por el MOOC, encuestas), se procede a emplear el Data Warehouse creado en el Capítulo 5 para poder realizar el análisis e interpretación de la información, con el fin de poder medir la eficacia del uso de los MOOCs en la educación.

## 6.7 Análisis de datos

En esta sección se busca responder las preguntas generadas en este caso de estudio y realizar un análisis mediante Data Warehouse

### 6.7.1 Respuesta a preguntas del caso de estudio

- ¿Cuáles son los estilos de aprendizaje predominantes dentro de los estudiantes de cada muestra?

NombreDocente	NombreEstilo1	CantidadAlumnos	
		NUEVO	REPETIDOR
DOCENTE 1	ACTIVO		7
	PRAGMATICO	4	
	REFLEXIVO	5	5
	TEORICO	1	3
DOCENTE 2	ACTIVO	11	3
	PRAGMATICO	7	
	REFLEXIVO	8	2
	TEORICO	1	2
DOCENTE 3	ACTIVO	2	1
	PRAGMATICO	3	3
	REFLEXIVO	9	2
	TEORICO	3	1
DOCENTE 4	ACTIVO	3	4
	PRAGMATICO	2	3
	REFLEXIVO	9	5
	TEORICO	1	4
DOCENTE 5	ACTIVO	3	
	PRAGMATICO	7	1
	REFLEXIVO	6	5
	TEORICO	4	

Figura 6.2: Reporte Muestra-EstiloAprendizaje  
Fuente: Elaboración propia

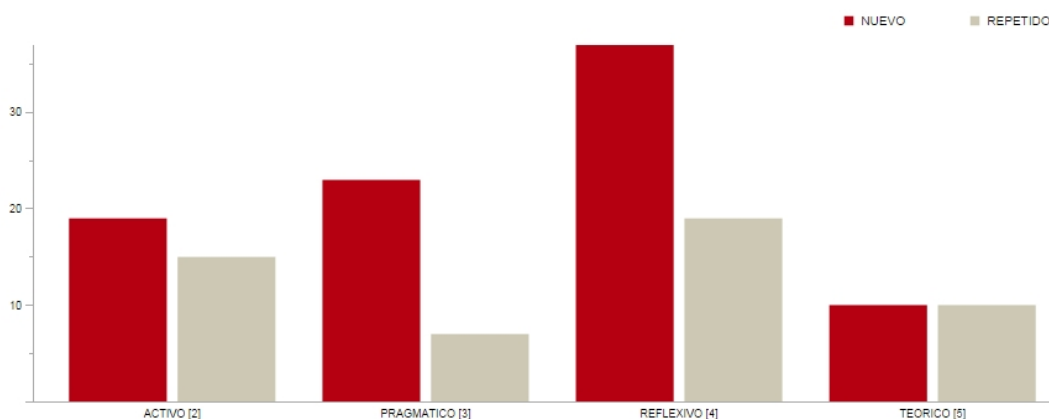


Figura 6.3: Gráfico del reporte Muestra-EstiloAprendizaje

Fuente: Elaboración propia

Como se puede observar en la *Figura 6.2*, para la muestra 1 el estilo de aprendizaje predominante de los estudiantes nuevos es el reflexivo, mientras que para los repetidores es el activo. Para la muestra 2 el estilo de aprendizaje predominante de los estudiantes nuevos y repetidores es el activo. Para la muestra 3 el estilo de aprendizaje predominante de los estudiantes nuevos es el reflexivo, mientras que de los estudiantes repetidores es el pragmático. Para la muestra 4 el estilo de aprendizaje predominante de los estudiantes nuevos y repetidores es el reflexivo. Para la muestra 5 el estilo de aprendizaje predominante de los estudiantes nuevos es el pragmático, mientras que de los estudiantes repetidores es el reflexivo.

En la *Figura 6.3*, considerando todas las muestras, el nivel predominante en los estudiantes nuevos es el reflexivo, mientras que para los estudiantes repetidores es el pragmático.

- ¿Cuáles son los estilos de aprendizaje según el sexo dentro de los estudiantes de cada muestra?

Como se puede observar en la *Figura 6.4*, para los estudiantes de sexo masculino los estilos de aprendizaje predominantes son:

- Muestra 1:
  - Repetidores: Activos
  - Nuevos: Activos y Reflexivos
- Muestra 2:
  - Repetidores: Activo
  - Nuevos: Activo
- Muestra 3:
  - Repetidores: Pragmático y Reflexivo
  - Nuevos: Reflexivo
- Muestra 4:
  - Repetidores: Reflexivo
  - Nuevos: Activo, Pragmático y Reflexivo.



- Muestra 5:
  - Repetidores: Reflexivo
  - Nuevos: Pragmático

NombreDocente	NombreEstilo1	DescripcionSexo	CantidadAlumnos	
			NUEVO	REPETIDOR
DOCENTE 1	ACTIVO	FEMENINO		1
		MASCULINO		6
	PRAGMATICO	MASCULINO	4	
	REFLEXIVO	FEMENINO	1	1
		MASCULINO	4	4
	TEORICO	MASCULINO	1	3
DOCENTE 2	ACTIVO	FEMENINO	3	1
		MASCULINO	8	2
	PRAGMATICO	MASCULINO	7	
	REFLEXIVO	FEMENINO	3	1
		MASCULINO	5	1
	TEORICO	FEMENINO		1
DOCENTE 3	ACTIVO	FEMENINO		1
		MASCULINO	2	1
	PRAGMATICO	FEMENINO	1	1
		MASCULINO	2	2
	REFLEXIVO	FEMENINO	2	
		MASCULINO	7	2
DOCENTE 4	ACTIVO	FEMENINO		1
		MASCULINO	3	3
	PRAGMATICO	MASCULINO	2	3
	REFLEXIVO	FEMENINO	1	2
		MASCULINO	8	3
	TEORICO	FEMENINO	1	
DOCENTE 5	ACTIVO	FEMENINO		1
		MASCULINO	7	
	PRAGMATICO	FEMENINO		1
		MASCULINO	7	
	REFLEXIVO	FEMENINO	2	1
		MASCULINO	4	4
	TEORICO	FEMENINO	1	
		MASCULINO	3	

Figura 6.4: Reporte Muestra-EstiloAprendizaje-Sexo

Fuente: Elaboración propia

Para las estudiantes de sexo femenino los estilos de aprendizaje predominantes son:

- Muestra 1:
  - Repetidores: Activo y Reflexivo
  - Nuevos: Reflexivos
- Muestra 2:
  - Repetidores: Activo, Reflexivo y Teórico
  - Nuevos: Activo y Reflexivo
- Muestra 3:
  - Repetidores: Teórico, Pragmático y Activo
  - Nuevos: Reflexivo

- Muestra 4:
  - Repetidores: Reflexivo
  - Nuevos: Reflexivo y Teórico
- Muestra 5:
  - Repetidores: Reflexivo y Pragmático
  - Nuevos: Reflexivo
- ¿En qué muestra se obtienen los mejores resultados?

	BUENA	EXCELENTE	INSUFICIENTE	MUY BUENA	REGULAR
NombreDocente	CantidadAlumnos	CantidadAlumnos	CantidadAlumnos	CantidadAlumnos	CantidadAlumnos
DOCENTE 1	5		11	4	5
DOCENTE 2	8	2	6	2	16
DOCENTE 3	1		16	1	6
DOCENTE 4	8		7	9	7
DOCENTE 5	5	1	11	6	3

Figura 6.5: Reporte Muestra-Resultados

Fuente: Elaboración propia

Como se puede observar en la *Figura 6.5*, la muestra 2 es la que mejores resultados presenta puesto que tiene la mayor cantidad de estudiantes en el grupo de calificaciones excelentes y menos estudiantes en el grupo de calificaciones de insuficiente.

- ¿Cuáles son los estilos de aprendizaje predominantes en los estudiantes de mayor rendimiento de cada muestra?

	BUENA	EXCELENTE	INSUFICIENTE	MUY BUENA	REGULAR
NombreEstilo1	CantidadAlumnos	CantidadAlumnos	CantidadAlumnos	CantidadAlumnos	CantidadAlumnos
ACTIVO	6	1	11	6	10
PRAGMATICO	4	1	13	3	9
REFLEXIVO	10	1	20	10	15
TEORICO	7		7	3	3

Figura 6.6: Reporte EstiloAprendizaje-Evaluación

Fuente: Elaboración propia

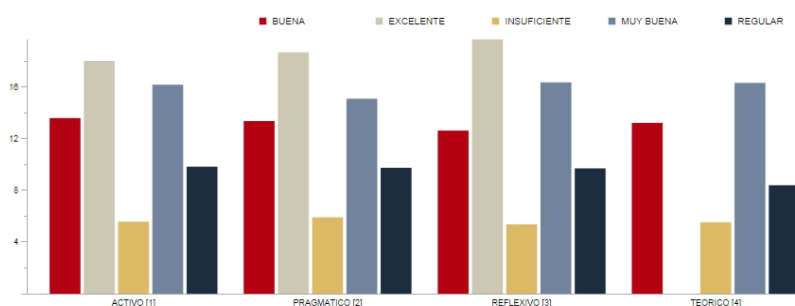


Figura 6.7: Gráfico reporte Estilo-Aprendizaje

Fuente: Elaboración propia

- ¿Cuál es el resultado obtenido por alumnos que repiten la materia o tienen conocimientos previos de programación contra los que son completamente nuevos en el tema?

DescripcionEstado	DescripcionCP	BUENA	EXCELENTE	INSUFICIENTE	MUY BUENA	REGULAR
		CantidadAlumnos	CantidadAlumnos	CantidadAlumnos	CantidadAlumnos	CantidadAlumnos
NUEVO	COLEGIO	5		7	7	9
	NO	6	2	30	5	18
REPETIDOR	COLEGIO	4		2	2	2
	INSTITUTOS	2		1		
	NO	10	1	11	8	8

Figura 6.8: Reporte Estado-ConocimientoPrevio

Fuente: Elaboración propia

Como se puede observar en la *Figura 6.8*, los estudiantes que son repetidores han obtenido mejores calificaciones que los estudiantes que son completamente nuevos, pero la diferencia no es significativa.

- De los alumnos que repiten la materia dependiendo de si siguieron o no el MOOC ¿quién obtuvo mejor media en lo teórico?  
De los alumnos que no repiten la materia dependiendo de si siguieron o no el MOOC ¿quién obtuvo mejor media en lo teórico?

DescripcionTA	EvaluacionTeorico	
	NUEVO	REPETIDOR
CON MOOC	6	6
SIN MOOC	6	6

Figura 6.9: Reporte Evaluación teórico

Fuente: Elaboración propia

Statistics	EvaluacionTeorico / NUEVO	EvaluacionTeorico / REPETIDOR
Min	5.612	5.652
Max	6.371	6.244
Sum	11.983	11.896
Average	5.992	5.948
Std. Deviation	0.379	0.296

Figura 6.10: Estadísticas del Reporte Evaluación Teórica

Fuente: Elaboración propia

Como se puede observar en la *Figura 6.9*, la media de los estudiantes repetidores y nuevos en la evaluación teórica de ambas muestras es la misma, tanto de los que tomaron el MOOC como de los que asistieron a clases presenciales.

- De los alumnos que repiten la materia dependiendo de si siguieron o no el MOOC ¿quién obtuvo mejor media en lo práctico?  
De los alumnos que no repiten la materia dependiendo de si siguieron o no el MOOC ¿quién obtuvo mejor media en lo práctico?

DescripcionTA	EvaluacionPractico	
	NUEVO	REPETIDOR
CON MOOC	4	5
SIN MOOC	3	6

Figura 6.11: Reporte Evaluación Práctico  
Fuente: Elaboración propia

Statistics	EvaluacionPractico / NUEVO	EvaluacionPractico / REPETIDOR
Min	3.038	4.500
Max	4.176	5.534
Sum	7.214	10.034
Average	3.607	5.017
Std. Deviation	0.569	0.517

Figura 6.12: Estadísticas del Reporte Evaluación Práctico  
Fuente: Elaboración propia

Como se puede observar en la *Figura 6.11*, la mejor media de los estudiantes repetidores en la evaluación práctica es la de los estudiantes que no tomaron el MOOC; mientras que la mejor media de los estudiantes nuevos en la evaluación práctica es la de los estudiantes tomaron el MOOC.

- ¿Cuál es el resultado de la participación de los alumnos en el MOOC según los estilos de aprendizaje?

NombreEstilo1	APROBADO	DESERTADO	NO APLICA
	CantidadAlumnos	CantidadAlumnos	CantidadAlumnos
ACTIVO	17	4	13
PRAGMATICO	9	2	19
REFLEXIVO	15	5	36
TEORICO	5	2	13

Figura 6.13: Reporte EstiloAprendizaje-ResultadoMooc  
Fuente: Elaboración propia

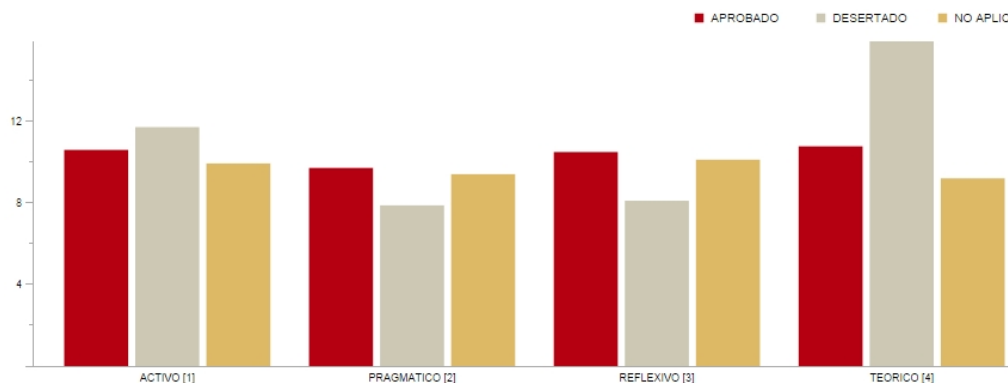


Figura 6.14: Gráfico EstiloAprendizaje-ResultadoMooc  
Fuente: Elaboración propia

Como se puede observar en la Figura 6.12, según los estilos de aprendizaje se puede decir que:

- De los activos 17 estudiantes aprobaron el MOOC y 4 desertaron.
  - De los pragmáticos 9 estudiantes aprobaron el MOOC y 2 desertaron.
  - De los reflexivos 15 estudiantes aprobaron el MOOC y 5 desertaron.
  - De los teóricos 5 estudiantes aprobaron el MOOC y 2 desertaron.
- ¿Cuál es el promedio del total de las calificaciones obtenidas por los alumnos en la evaluación según los estilos de aprendizaje?

NombreEstilo1	EvaluacionGlobal	EvaluacionPractico	EvaluacionTeorico
ACTIVO	10	5	6
PRAGMATICO	9	3	6
REFLEXIVO	10	4	6
TEORICO	10	4	6

Figura 6.15: Reporte EstiloAprendizaje-Evaluación  
Fuente: Elaboración propia

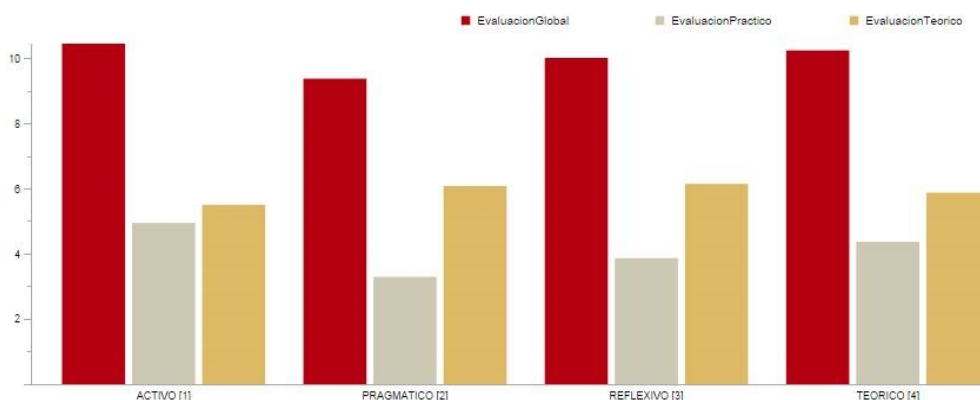


Figura 6.16: Gráfico Reporte EstiloAprendizaje-Evaluación  
Fuente: Elaboración propia

En la *Figura 6.14*, se puede observar que en la evaluación global los estilos de aprendizaje activo, reflexivo y pragmático obtienen mejor promedio que el pragmático; en la evaluación práctica el estilo de aprendizaje que obtiene mejor promedio es el activo, mientras que en la evaluación teórica todos los estilos de aprendizaje tienen el mismo promedio.

## 6.7.2 Aplicación de minería de datos educacional

En esta sección se aplica Minería de Datos mediante el modelo conceptual que se obtuvo en el anterior capítulo y se utiliza los datos de este caso de estudio.

### 6.7.2.1 Clasificación de un tipo de estudiante

#### Definición del problema

**Qué tienen en común los estudiantes de una determinada ciudad, que tengan o no conocimiento previo sobre programación, en un rango de edad y con diferentes estilos de aprendizaje.**

Este análisis se enfoca en clasificar un tipo de estudiantes, en este caso un estudiante de una determinada ciudad, con o sin conocimiento previo sobre programación, en un rango de edad específico y con diferentes estilos de aprendizaje. A este tipo de estudiantes se les clasificará en excelente, muy bueno, bueno, regular e insuficiente en base a las calificaciones obtenidas y a los parámetros ingresados.

#### Elección del algoritmo

El problema planteado intenta obtener como resultado la clasificación de la notas obtenidas por los estudiantes de la asignatura de Algoritmos, Datos y Estructuras ofertada por la Facultad de Ingeniería de la Universidad de Cuenca en los tres primeros capítulos del sílabo. Este análisis intentará descubrir el patrón más adecuado que se ajuste con a los resultados obtenidos previamente, por lo que se utilizará el algoritmo más común en clasificación que el de NativeBayes y el J48.

#### Obtención de Datos

Para la obtención de los datos se define la consulta SQL que se muestra en la *Figura 6.16*, en la cual los datos son extraídos de la base de datos del modelo multidimensional.

```
SELECT ciudad.nombre,conocimientoprevio.Descripcion as conocimientoprevio,docente.Nombre as docente,edad.Descripcion as edad,
estiloprimary.Descripcion as estilo1,estilosecundario.Descripcion as estilo2,sexo.Descripcion as sexo,tipoestudiante.Descripcion as tipoestudiante,
resultadoevaluacion.Descripcion as resultado
FROM hechos,ciudad,conocimientoprevio,docente,edad,estiloprimary,estilosecundario,sexo,tipoestudiante,resultadoevaluacion
where hechos.idCiudad=ciudad.idCiudad and hechos.idConocimientoPrevio=conocimientoprevio.idConocimientoPrevio
and hechos.idDocente=docente.idDocente and hechos.idEdad=edad.idEdad and hechos.idEstiloPrimario= estiloprimary.idEstiloPrimario
and hechos.idEstiloSecundario=estilosecundario.idEstiloSecundario and hechos.idSexo=sexo.idSexo and
hechos.idTipoEstudiante=tipoestudiante.idTipoEstudiante and hechos.idResultado=resultadoevaluacion.idResultado;
```

*Figura 6.17: Consulta SQL de los datos de entrada*  
Fuente: Elaboración propia

Como se puede observar en la consulta existen ocho variables a analizar:

Variable	Descripción
<b>Ciudad</b>	Variable que almacena la ciudad de procedencia del estudiante.
<b>Conocimiento Previo</b>	Variable que almacena si el estudiante tenía o no conocimientos previos programación antes de tomar la asignatura de Algoritmos, Datos y Estructuras.

<b>Docente</b>	Variable que almacena el docente con el cuál el estudiante tomó la asignatura de Algoritmos, Datos y Estructuras.
<b>Edad</b>	Variable que almacena la edad del estudiante.
<b>Estilo primario</b>	Variable que almacena el estilo de aprendizaje predominante en el estudiante.
<b>Estilo secundario</b>	Variable que almacena el estilo de aprendizaje secundario en el estudiantes
<b>Sexo</b>	Variable que almacena el sexo del estudiante.
<b>Tipo Estudiante</b>	Variable que almacena si el estudiante curso o no el MOOC.

Tabla 6.1: Variables de entrada para clasificación.

Fuente: Elaboración propia

## Minería de datos

Las características principales del modelo son:

- **Algoritmo a utilizar:** NaiveBayes y J48
- **Objetivo del modelo:** Clasificar al estudiante de la asignatura de Algoritmos, Datos y Estructuras en base a las notas obtenidas.
- **Atributos a predecir:** ciudad, conocimiento previo, docente, edad, estilo primario, estilo secundario, sexo, tipo estudiante.

## Carga de datos

Una vez que se han identificado los atributos y después de tener codificada la consulta SQL se procede a la aplicación y ejecución del algoritmo seleccionado.

El primer paso a realizar es la conexión de WEKA con la base de datos, este procedimiento se puede observar en la *Figura 6.17*.

## Aplicación del Algoritmo: NaiveBayes

Una vez establecida la conexión se procede con la selección del algoritmo y la definición de los respectivos atributos, tal como se puede observar en la *Figura 6.18*.

La *Figura 6.19* presenta la clasificación utilizando el algoritmo de NaiveBayes. En la misma se puede apreciar la clasificación realizada considerando las variables del estilo de aprendizaje predominante y el tipo de estudiante.

Por ejemplo, si un estudiante es reflexivo, según la *Figura 6.20*, se lo ubicaría directamente en el grupo de estudiantes con calificación de *Buena*.

En el otro caso, si el estudiante tomó el MOOC se lo ubicaría directamente en el grupo de estudiantes con calificación *Regular*, mientras que si no tomó el MOOC iría al grupo de *Insuficiente*.

En la *Figura 6.21*, se puede observar que las instancias clasificadas correctamente por medio del algoritmo de NaiveBayes corresponden al 67,8571%.



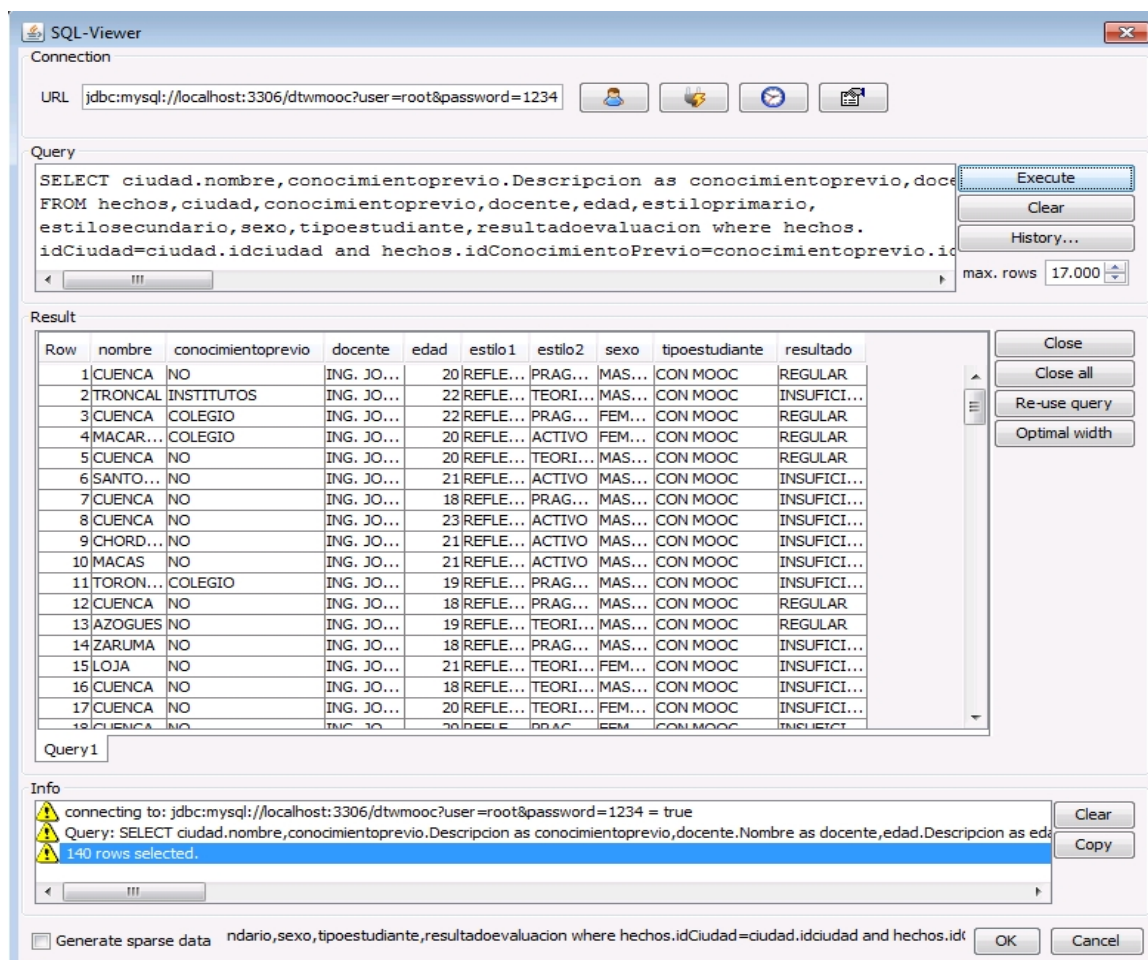


Figura 6.18: Carga de datos en Weka  
Fuente: Elaboración propia

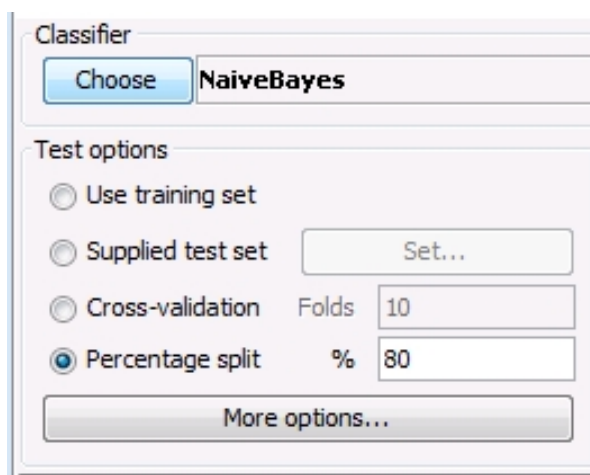


Figura 6.19: Configuración del algoritmo NaiveBayes  
Fuente: Elaboración propia

Attribute	Class				
	BUENA (0.19)	INSUFICIENTE (0.36)	REGULAR (0.26)	MUY BUENA (0.16)	EXCELENTE (0.03)
<b>estilo1</b>					
PRAGMATICO	11.0	9.0	11.0	10.0	2.0
TEORICO	7.0	21.0	14.0	6.0	2.0
ACTIVO	5.0	18.0	9.0	5.0	2.0
REFLEXIVO	8.0	7.0	7.0	5.0	1.0
[total]	31.0	55.0	41.0	26.0	7.0
<b>tipoestudiante</b>					
CON MOOC	14.0	18.0	22.0	7.0	3.0
SIN MOOC	15.0	35.0	17.0	17.0	2.0
[total]	29.0	53.0	39.0	24.0	5.0

Figura 6.20: Resultado de datos analizados  
Fuente: Elaboración propia

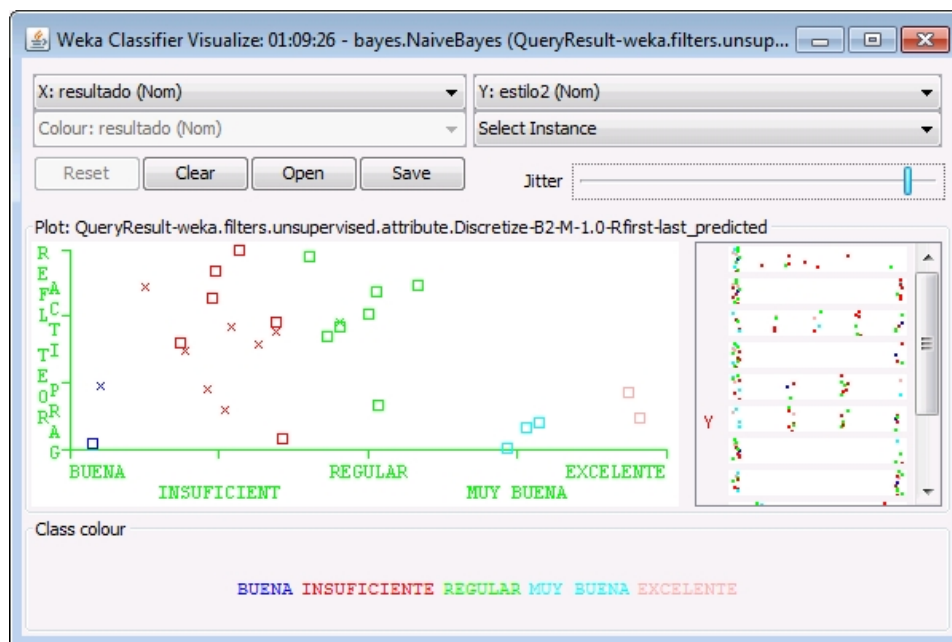


Figura 6.21: Gráfico de datos analizados  
Fuente: Elaboración propia

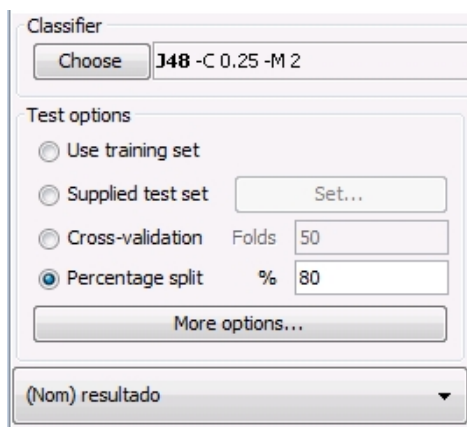
=== Summary ===

Correctly Classified Instances	19	67.8571 %
Incorrectly Classified Instances	9	32.1429 %
Kappa statistic	0.125	
Mean absolute error	0.2398	
Root mean squared error	0.3525	
Relative absolute error	91.4007 %	
Root relative squared error	102.483 %	
Coverage of cases (0.95 level)	89.2857 %	
Mean rel. region size (0.95 level)	68.75 %	
Total Number of Instances	28	

Figura 6.22: Resumen después de aplicar el algoritmo NaiveBayes  
Fuente: Elaboración propia

### Aplicación del Algoritmo: J48

Una vez establecida la conexión se procede con la selección del algoritmo y la definición de los respectivos atributos, tal como se puede observar en la *Figura 6.22*

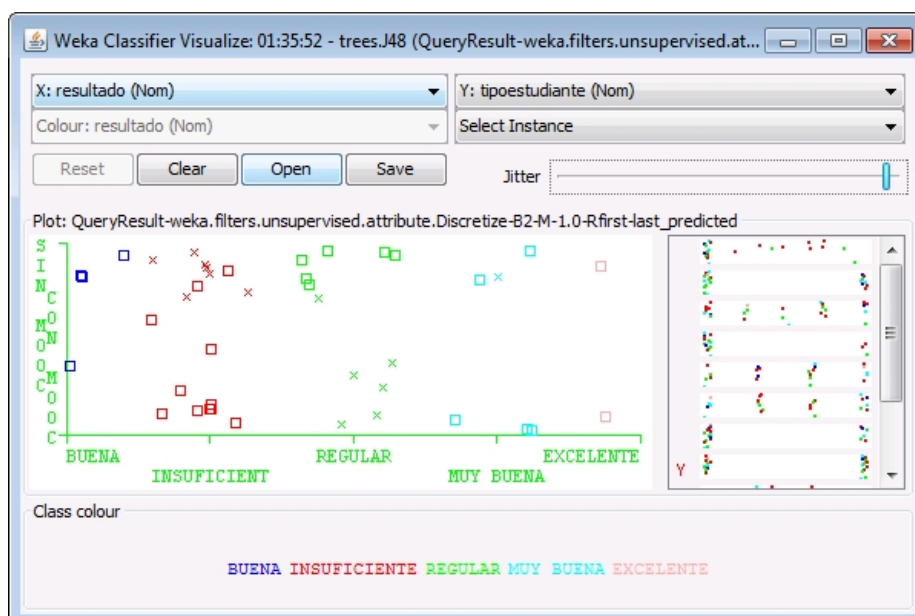


*Figura 6.23: Configuración del algoritmo J48*  
Fuente: Elaboración propia

La *Figura 6.23* presenta la clasificación utilizando el algoritmo J48. En la misma se puede apreciar la clasificación realizada considerando las variables del estilo de aprendizaje predominante y el tipo de estudiante.

De acuerdo al tipo de aprendizaje, podemos decir que si el estudiante es de tipo reflexivo, se lo ubicaría en el grupo de *Insuficiente*.

En el otro caso, si el estudiante tomó el MOOC se lo ubicaría directamente en el grupo de estudiantes con calificación *Regular*, mientras que si no tomó en MOOC iría al grupo de *Insuficiente*, tal como se lo puede observar en la *Figura 6.23*:



*Figura 6.24: Gráfico de datos analizados*  
Fuente: Elaboración propia

En la Figura 6.25, se puede observar que las instancias clasificadas correctamente por medio del algoritmo J48 corresponden al 75% del total.

```

=== Summary ===

Correctly Classified Instances      21          75    %
Incorrectly Classified Instances    7           25    %
Kappa statistic                     0
Mean absolute error                 0.2583
Root mean squared error             0.3427
Relative absolute error             98.4658 %
Root relative squared error         99.6343 %
Coverage of cases (0.95 level)     92.8571 %
Mean rel. region size (0.95 level) 75          %
Total Number of Instances          28

```

Figura 6.25: Resumen después de aplicar el algoritmo J48  
Fuente: Elaboración propia

### Comparación de resultados

Realizado el proceso de clasificación tanto con el algoritmo de NaiveBayes como con el algoritmo J48 se ha podido llegar a la conclusión que para el conjunto de datos analizado el algoritmo J48 presenta un error menor (25%) en las instancias clasificadas que el algoritmo de NaiveBayes (32,1429%), pero que a pesar de esta diferencia los resultados en la clasificación son iguales en ambos algoritmos.

### 6.7.2.2 Clúster de un tipo de estudiante

#### Definición del problema

**Qué características tienen en común los estudiantes de la asignatura de Algoritmos, Datos y Estructuras que tomaron el MOOC con los estudiantes que asistieron a las clases presenciales.**

Este análisis se enfoca agrupar o clusterizar a los estudiantes con un estilo de aprendizaje predominante, de acuerdo al sexo, a la forma en la que tomaron la asignatura (MOOC o presencial) y en base a la calificación obtenida (Excelente, muy buena, buena, regular, insuficiente).

#### Elección del algoritmo

El problema planteado intenta obtener como resultado patrones de clasificación de los estudiantes de la asignatura de Algoritmos, Datos y Estructuras que han tomado la asignatura ya sea de forma presencial o por medio del MOOC. Para esto se tomarán en consideración los algoritmos *FarthestFirst* y *MakeDensityBasedClusterer*.

#### Obtención de Datos

Para la obtención de los datos se define la consulta SQL que se muestra en la Figura 6.26, en la cual los datos son extraídos de la base de datos del modelo multidimensional.

```

SELECT estiloprimario.NombreEstilo1 as estilo1,sexo.DescripcionSexo as sexo,tipoestudiante.DescripcionTA as tipoestudiante,
resultadoevaluacion.DescripcionREG as resultado
FROM hechos,ciudad,conocimientoprevio,edad,estiloprimario,estilossecundario,sexo,tipoestudiante,resultadoevaluacion
where hechos.idConocimientoPrevio=conocimientoprevio.idConocimientoPrevio
and hechos.idEdad=edad.idEdad and hechos.idEstiloPrimario= estiloprimario.idEstiloPrimario and
hechos.idSexo=sexo.idSexo and hechos.idTipoEstudiante=tipoestudiante.idTipoEstudiante
and hechos.idResultado=resultadoevaluacion.idResultado;

```

Figura 6.26: Consulta SQL de los datos de entrada  
Fuente: Elaboración propia

Como se puede observar en la consulta existen cuatro variables a analizar:

Variable	Descripción
<b>Estilo primario</b>	Variable que almacena el estilo de aprendizaje predominante en el estudiante.
<b>Resultado</b>	Variable que almacena el grupo al que pertenece el estudiante según su calificación
<b>Sexo</b>	Variable que almacena el sexo del estudiante.
<b>Tipo Estudiante</b>	Variable que almacena si el estudiante curso o no el MOOC.

Tabla 6.2: Variables consideradas para clusterización

Fuente: Elaboración propia

## Minería de datos

Las características principales del modelo son:

- **Algoritmo a utilizar:** FarthestFirst y MakeDensityBasedClusterer
- **Objetivo del modelo:** Obtener como resultado patrones de clasificación de los estudiantes de la asignatura de Algoritmos datos y estructuras de la Universidad de Cuenca.
- **Atributos a predecir:** Estilo de aprendizaje, sexo, tipo de estudiante, resultado

## Carga de datos

Una vez que se han identificado los atributos y después de tener codificada la consulta SQL se procede a la aplicación y ejecución del algoritmo seleccionado.

El primer paso a realizar es la conexión de WEKA con la base de datos, este procedimiento se puede observar en la siguiente figura:

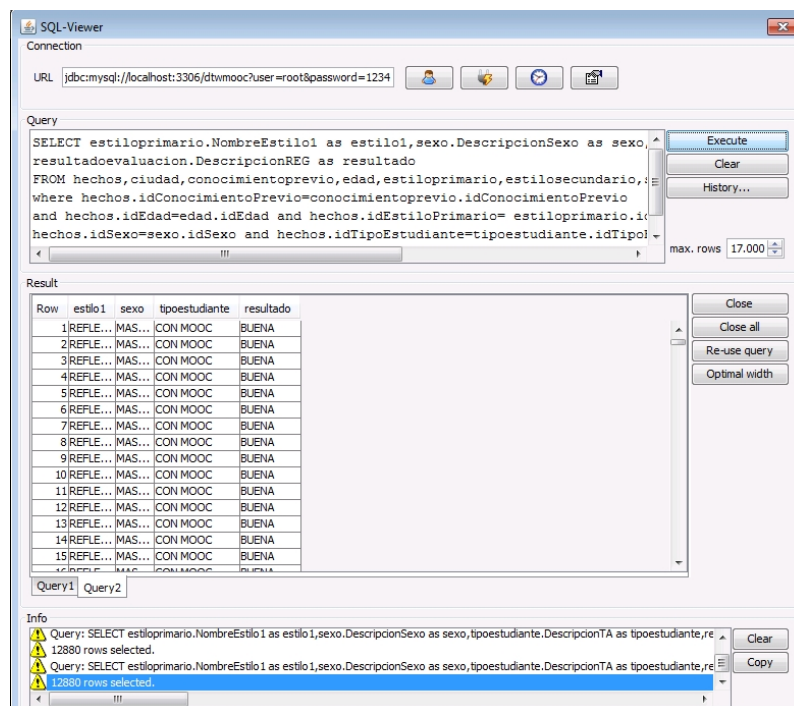


Figura 6.27: Carga de datos en el Weka

Fuente: Elaboración propia



### Aplicación del Algoritmo: FarthestFirst

Una vez establecida la conexión se procede a la selección del algoritmo, como se indica en la Figura 6.28.

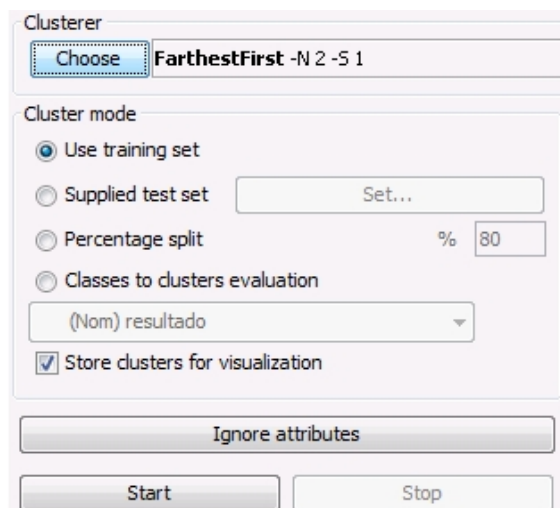


Figura 6.28: Configuración del algoritmo FarthestFirst  
Fuente: Elaboración propia

Realizado este proceso, en la Figura 6.29 se pueden apreciar los dos clúster generados utilizando este algoritmo.

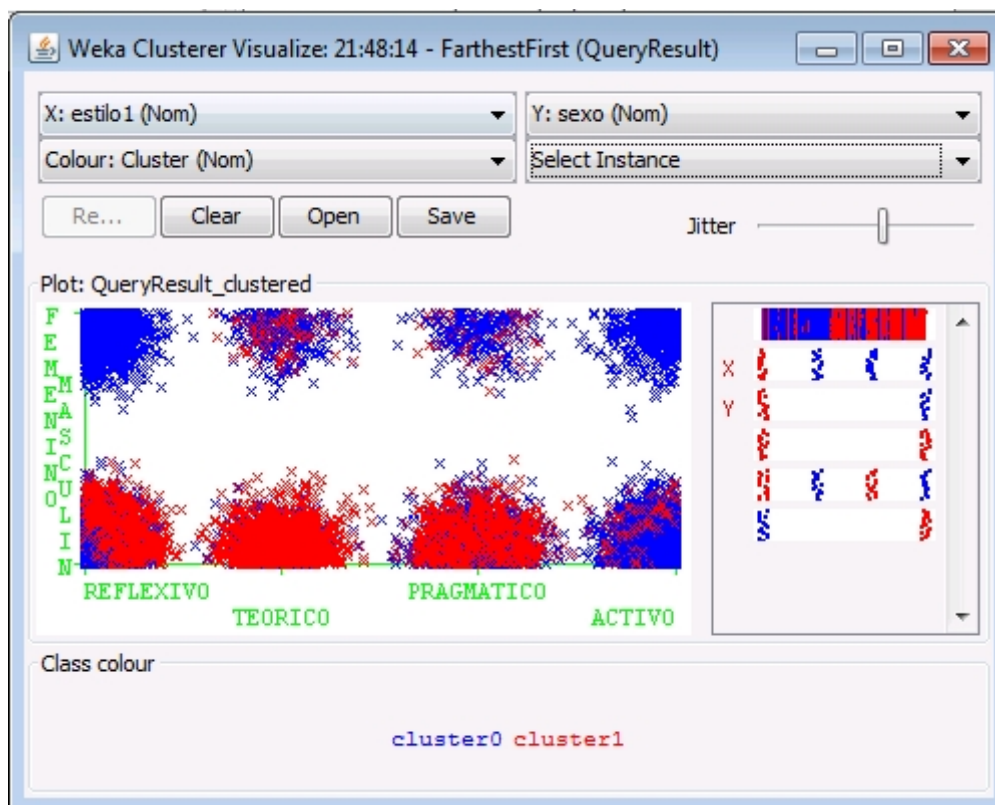


Figura 6.29: Gráfico de clusterización con el algoritmo FarthestFirst  
Fuente: Elaboración propia

Como se puede observar en el cluster 1 se agrupan la mayoría de estudiantes de sexo masculino con estilo de aprendizaje reflexivo, teórico y pragmático, mientras que el cluster 0 agrupa a los estudiantes de sexo femenino con estilo de aprendizaje reflexivo y activo.

En la Figura 6.30, se puede observar como WEKA establece los grupos para este algoritmo.

```

FarthestFirst
=====

Cluster centroids:

Cluster 0
    ACTIVO FEMENINO CON MOOC INSUFICIENTE
Cluster 1
    TEORICO MASCULINO SIN MOOC REGULAR

Time taken to build model (full training data) : 0.02 seconds

=== Model and evaluation on training set ===

Clustered Instances

0          6532 ( 51%)
1          6348 ( 49%)

```

Figura 6.30: Resumen después de aplicar el algoritmo FarthestFirst

Fuente: Elaboración propia

### Aplicación del Algoritmo: MakeDensityBasedClustered

Una vez establecida la conexión se procede a la selección del algoritmo, como se indica en la Figura 6.31.

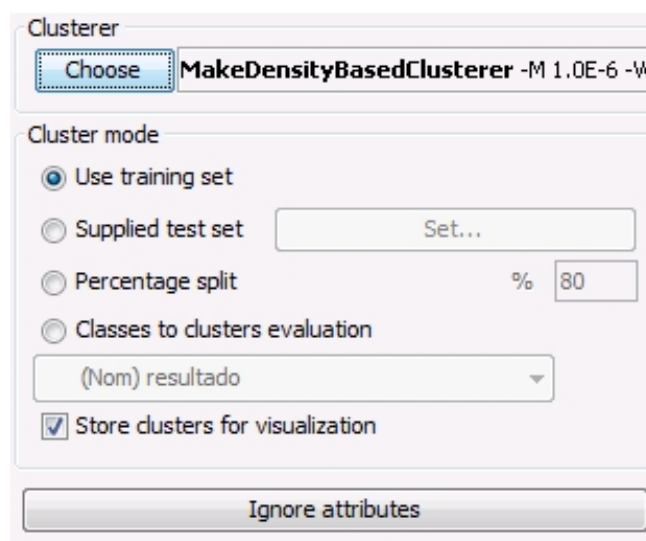


Figura 6.31: Configuración del algoritmo MakeDensityBasedClustered

Fuente: Elaboración propia



Realizado este proceso, en la Figura 6.32 se pueden apreciar los dos clúster generados utilizando este algoritmo.

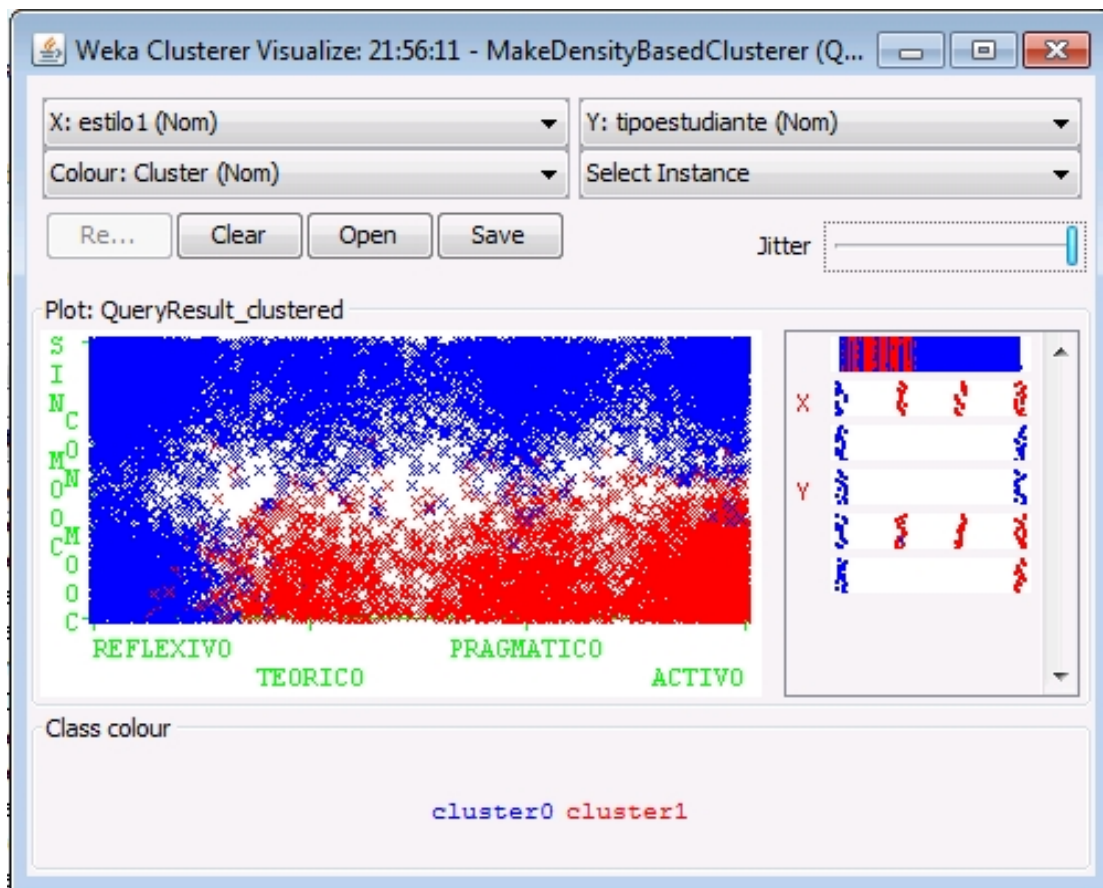


Figura 6.32: Resultado después de aplicar el algoritmo MakeDensityBasedClusterer  
Fuente: Elaboración propia

Como se puede observar en el clúster 1 se agrupan la mayoría de estudiantes que tomaron el MOOC y que pertenecen al estilo de aprendizaje teórico, pragmático y activo; mientras que en el clúster 0 se agrupan los estudiantes que no tomaron el MOOC con estilos de aprendizaje reflexivo, teórico, pragmático y activo, y los estudiantes que tomaron el MOOC con estilo de aprendizaje reflexivo.

En la Figura 6.33, se puede observar como WEKA establece los grupos para este algoritmo.

## 6.8 Interpretación de Resultados

Una vez que se han analizado los datos obtenidos en el caso de estudio se puede decir que:

- El estilo de aprendizaje predominante en los estudiantes nuevos es el reflexivo, mientras que en los estudiantes repetidores es el activo.
- La muestra del Docente 2 que utilizó el MOOC ha obtenido los mejores resultados generales, puesto que es la que más alumnos aportó al grupo de calificaciones excelentes, y la que menos alumnos tiene en el grupo de calificaciones insuficientes.



```

Cluster: 0 Prior probability: 0.7214

Attribute: estilo1
Discrete Estimator. Counts = 5153 1197 1749 1197 (Total = 9296)
Attribute: sexo
Discrete Estimator. Counts = 7361 1933 (Total = 9294)
Attribute: tipoestudiante
Discrete Estimator. Counts = 1841 7453 (Total = 9294)
Attribute: resultado
Discrete Estimator. Counts = 1841 5797 1565 93 (Total = 9296)

Cluster: 1 Prior probability: 0.2786

Attribute: estilo1
Discrete Estimator. Counts = 1 645 1013 1933 (Total = 3592)
Attribute: sexo
Discrete Estimator. Counts = 3037 553 (Total = 3590)
Attribute: tipoestudiante
Discrete Estimator. Counts = 3589 1 (Total = 3590)
Attribute: resultado
Discrete Estimator. Counts = 645 2301 461 185 (Total = 3592)

Time taken to build model (full training data) : 0.24 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      9292 ( 72%)
1      3588 ( 28%)

```

Figura 6.33: Resumen después de aplicar el algoritmo *MakeDensityBasedClusterer*  
Fuente: Elaboración propia

- Los conocimientos previos que tengan los estudiantes sobre programación antes de iniciar la asignatura de Algoritmos, Datos y Estructuras no es una ventaja sobre aquellos estudiantes que no hayan tenido conocimientos de programación, ya que en base a los resultados obtenidos, los alumnos sin conocimientos previos muestran mejores calificaciones incluso sobre aquellos estudiantes que están repitiendo la asignatura.
- Los estudiantes que no completaron el MOOC representan el 13,55%, siendo los estudiantes que si completaron el MOOC el 86,45%, lo cual representa una mayoría considerable.
- En la parte teórica, el promedio de notas obtenidas por los estudiantes que cursaron el MOOC es igual al promedio de los estudiantes que asistieron a clases presenciales, este promedio es de 6 sobre 10.
- En la parte práctica, el promedio de notas obtenidas por los estudiantes que cursaron el MOOC es mayor en los estudiantes nuevos cuyo promedio es 4 sobre 10, mientras que en los estudiantes que no cursaron el MOOC el promedio es mayor en los estudiantes que están repitiendo la asignatura cuyo promedio es de 6 sobre 10.
- El MOOC resultó más beneficioso para estudiantes con estilos de aprendizaje Activo y Pragmático, puesto que fueron los que mejores calificaciones obtuvieron en la evaluación final, mientras que los estudiantes con estilo de Aprendizaje Reflexivo y Teórico obtuvieron mejores calificaciones asistiendo a las clases magistrales.



## 6.9 Recapitulación

En este capítulo se realizó un caso de estudio basado en la asignatura de Algoritmos, Datos y Estructuras, dentro de este caso de estudio se manejaron cinco muestras en donde dos de estas emplearon el MOOC generado como parte del Capítulo 4, mientras que las otras tres muestras llevaban las clases de forma tradicional. Como medio para poder realizar los análisis y la interpretación de los datos respectivos se utilizó el Data Warehouse creado en el Capítulo 5, así como también se empleó ese mismo modelo conceptual de datos para aplicar minería de datos educacional, en donde se pudo observar que debido a la corta cantidad de datos los algoritmos mostraron altos porcentajes de error. Como resultado de este caso de estudio se puede resumir que el MOOC sirvió de ayuda a los estudiantes y que debería ser considerado para aplicaciones futuras. Mientras que por otra parte el estilo de aprendizaje predominante en alumnos nuevos es el reflexivo por lo que debería considerarse este parámetro dentro de las metodologías de enseñanza utilizadas por los docentes de la Facultad de Ingeniería de la Universidad de Cuenca.



# CAPÍTULO 7:

## Conclusiones y trabajos futuros

### 7.1 Conclusiones

En este trabajo se demuestra la gran utilidad de los MOOCs en el ámbito académico y su influencia en el proceso de aprendizaje de los alumnos. Las conclusiones generadas mediante este proyecto de tesis se exponen a continuación:

- Los cursos en línea, masivos y abiertos han evolucionado desde su aparición en el año 2008 en el que fueron creados generando un nuevo paradigma en la educación online. Además de contar con características muy importantes como la modularidad, escalabilidad y capacidad de recombinación tienen la ventaja de ser gratuitos, estar disponibles las 24 horas del día los 7 días a la semana, lo que facilita el acceso a los mismos.
- En el mercado existen varias plataformas que ofertan MOOCs al público entre las más conocidas tenemos a Google Course Builder, EdX, Coursera, MiriadaX, Udacity. Cada una de ellas posee distintas características que las diferencian unas de otras, siendo Google Course Builder la que se utilizó para el desarrollo de esta tesis, las razones por las que se escogió esta plataforma son: ofrece un servidor gratuito para la publicación del MOOC, permite el uso de todas las aplicaciones que brinda Google para el desarrollo del mismo, permite traducir el MOOC a más de 50 idiomas con sólo cambiar la configuración del archivo principal, permite exportar el MOOC para que pueda ser montado sobre otro servidor web con características similares a las que ofrece Google.
- La herramienta de Google Course Builder utilizada para la implementación del MOOC fue de gran utilidad puesto que resultó ser una herramienta que en base a HTML, CSS y JavaScript permitía ir configurando cada una de las opciones para crear el MOOC. La plataforma permite además modificar el código de manera que se pueda crear otra estructura con un diseño diferente del MOOC, aunque para estas opciones se requería de conocimientos en Python para poder modificar los scripts que posee esta herramienta.



- El modelo conceptual de datos obtenido en el Capítulo 5 cubre muchos aspectos y es bastante general, por lo que podría servir de base para realizar estudios varios sobre la puesta en marcha de otros MOOCs y obtener información importante para poder tomar decisiones.
- Aunque el tiempo de prueba empleado para el despliegue y uso del MOOC fue corto, se pudo observar a través del análisis realizado mediante el modelo conceptual de datos que el MOOC permitió a los alumnos obtener calificaciones más aceptables que aquellos que solo llevaron las clases presenciales de forma tradicional sin el apoyo de recursos tecnológicos.

## 7.2 Líneas de trabajo futuras

El fenómeno de los MOOCs ha arrancado con mucha fuerza en diversas universidades en el mundo como España, Estados Unidos y diversos países de América Latina, esto ha permitido establecer alianzas con potencias tecnológicas de la importancia de Google para el desarrollo e implementación de estos nuevos tipos para el desarrollo del aprendizaje.

En el desarrollo de la presente tesis se ha realizado un prototipo de implementación de MOOC enfocado hacia los estudiantes de la Facultad de Ingeniería que cursaban la materia Algoritmos, Datos y Estructuras obteniendo resultados positivos en la aceptación de la herramienta para una mejor comprensión de la asignatura.

Los MOOCs serían excelentes oportunidades en el desarrollo del aprendizaje de los estudiantes, puesto que se trata de una herramienta que contiene diferentes materiales educativos digitales por medio de los cuales se puede afianzar el conocimiento adquirido en las aulas de clase. Esta herramienta puede ser aplicada a diversas asignaturas de múltiples carreras que busquen innovar en su sistema educativo y permita acoplarse a las nuevas herramientas tecnológicas con las cuales se cuenta.

La movilidad y acceso al Internet ha fijado una nueva frontera en la carrera tecnológica en la cual nos encontramos en la actualidad, a nivel educativo esta se ha visto fomentada por la educación a distancia obteniendo resultados positivos de tal forma que se ha incrementado las opciones para el desarrollo educativo en los últimos tiempos. Conocedores de esta realidad, Google ha generado la herramienta de Google Course Builder para el diseño e implementación de MOOCs. La Universidad de Cuenca puede emplear estas ventajas y convertirse en un partner de Google e implementar un laboratorio especializado en desarrollo de objetos de aprendizaje y ofertar cursos para sus estudiantes, tanto escolarizados como externos, en calidad de educación continua.

La modalidad de este tipo de cursos respecto a la duración y a las horas de dedicación respetaría una media, que se ha acoplado internacionalmente, de 4 a 7 semanas con una dedicación estimada de 4 horas por semana. Las temáticas que se han implementado a nivel internacional se han visto enfocada principalmente en Ciencias y Tecnología seguida de Ciencias Sociales y Jurídicas, esto denota los puntos hacia los cuales se podría extender la modalidad de cursos virtuales e implementación de MOOCs para líneas futuras.



Dentro de las políticas a adoptarse a partir del desarrollo de la presente tesis se encuentran las siguientes:

- Realizar un estudio detallado de la demanda de formación MOOC a partir de datos empíricos de cursos ya impartidos.
- Buscar un modelo que permita analizar el impacto de este tipo de formación en la sociedad, tanto desde el punto de vista de los planes formativos de las empresas como la formación ocupacional para colectivos sin empleo.
- Analizar y definir métricas que permitan realizar una evaluación del rendimiento de los participantes en los MOOC a partir de su nivel social.

La aplicación de estas políticas o líneas de trabajo futuras permitirán dar lugar a nuevas ediciones de MOOCs en el entorno local, pudiendo convertir a la Universidad de Cuenca en pionera en este tipo de metodologías, puesto que existen instituciones como la Universidad Técnica Particular de Loja que ya se encuentra trabajando en estas nuevas líneas de desarrollo educacional.

Acorde a las encuestas de satisfacción acerca del MOOC realizadas, se han obtenido las siguientes observaciones a tomar en cuenta para un mejor desarrollo en el futuro:

- En el acceso con dispositivos móviles se deberán realizar mayor cantidad de pruebas y considerar la realización de una aplicación móvil nativa tanto para Android como para iOS y Windows Phone.
- Mejorar la calidad de los videos que se presentan, personalizándolos y diseñándolos de una manera mucho más didáctica y explicativa para los estudiantes.
- Generar contenidos mucho más básicos, en primera instancia se denotó que existía la necesidad de revisar más de una vez los contenidos, esto puede ser mejorado generando conceptos más cortos y específicos.
- Incentivar a un mayor uso de los foros buscando una mayor participación de los estudiantes con incentivos que inviten a los participantes a interactuar con los contenidos.
- Generar mayor diversidad en los ejercicios a realizar para que los estudiantes puedan complementar de mejor manera los conceptos adquiridos al momento de desarrollar el MOOC.
- Mejorar la calidad de los materiales externos que se presenten para el desarrollo del MOOC, este debe ser más claro y específico y no superior a las tres páginas puesto que se desvirtúa el concepto didáctico que los MOOC plantean.

Por otra parte se podría emplear el modelo conceptual de datos generado en la presente tesis para poder analizar múltiples casos, es decir, se podría agregarse las dimensiones “Tiempo” y “Asignatura”, de tal manera que si se aplican MOOCs en múltiples asignaturas en diferentes períodos lectivos, entonces la cantidad de datos va a ser grande y los márgenes de error lanzados en la minería de datos educacional podrían reducirse significativamente, a tal punto que podrían generarse predicciones que permitan tomar decisiones importantes.

Como se puede apreciar en lo explicado a lo largo de esta sección, aún quedan muchas líneas de acción en las cuales se pueden trabajar a futuro para mejorar la implementación de MOOCs en la Universidad de Cuenca, se considera que el desarrollo de este primer acercamiento ha sido todo un éxito puesto que el nivel de aceptación del estudiantado ha sido amplio y nos permite determinar que resultaría ser un proyecto robusto, incremental y productivo en el cuál se puede innovar de muchas maneras.



## 7.3 Lecciones Aprendidas

- La búsqueda de información puede ser demasiado extensa si es que no se delimita que es lo que se está buscando exactamente, es por eso que se debe tomar en cuenta los procesos de búsqueda de información a seguir y cuáles son los resultados esperados.
- La selección de una plataforma para desplegar un MOOC debe realizarse en base a las necesidades que se tengan para realizar el mismo; cada plataforma tiene características propias que las diferencian unas de otras, por lo que se debe analizar que herramientas tecnológicas que utilizan cada una de ellas y que materiales educativos digitales admiten para la creación del MOOC. Por otra parte, también se debe considerar el alcance que va a tener el MOOC, puesto que en base a ello se debe verificar cuales son los recursos de hardware y software que necesita cada una de las plataformas para la implementación del mismo.
- La implementación de un MOOC en cuanto a plataformas, es un proceso que puede terminar siendo un problema si no se tiene claro que herramientas posee la plataforma y las tecnologías necesarias para poder crear el MOOC, es por esta razón que se debe seleccionar una plataforma que emplee tecnologías de nuestro dominio.
- La implementación de un MOOC en cuanto a contenidos, puede convertirse en un proceso que no presente mayores inconvenientes, siempre y cuando previamente se halla empleado una metodología de diseño que sirva de bitácora para ir construyendo el MOOC paso a paso.
- El diseño de un modelo conceptual de datos es un procedimiento que puede conllevar mucho tiempo para su desarrollo y del correcto análisis dependerá el éxito del mismo. Si el modelo obtenido es óptimo puede ser usado para múltiples aplicaciones.
- El desarrollo de un caso de estudio puede aportarnos con información valiosa, siempre y cuando el caso de estudio tenga sus distintos parámetros definidos claramente





# Referencias

- Aguilar Mayorga, S. M., & Lemus Castiblanco, J. L. (2009). *Pentaho-BI*. Tesis Pregrado, Universidad Nacional Colombia, Unidad de Informática y Comunicaciones, Bogotá. Recuperado el 31 de Agosto de 2014, de <http://www.fce.unal.edu.co/wiki/images/d/d9/Pentaho-BI.pdf>
- Alonso, & Gallego. (Julio de 2004). Los estilos de Aprendizaje. Una propuesta pedagógica. Chile.
- Bartolomé, A. (03 de 2014). MOOC: 4+2 años de expectativas y resultados.
- Bernabeu, R. (19 de Julio de 2010). *HEFESTO Data warehousing: investigación y sistematización de conceptos. HEFESTO: Metodología para la construcción de un Data warehouse*. Córdoba, Argentina. Recuperado el 10 de Septiembre de 2014
- Billings, D. M. (2014). Understanding Massively Open Online Courses. *The Journal of Continuing Education in Nursing*, 58-59.
- Bonvillian, W., & Singer, S. (2013). The Online Challenge to Higher Education. *Summer*, 23-30.
- Botero Villalba, L. F., & Mera, J. A. (2011). Modelo para evaluación y selección de software integrado de gestión para las instituciones de educación superior basado en estándares reconocidos internacionalmente. 91. Santiago de Calí, Colombia. Recuperado el Agosto de 2014
- Campos, O. (8 de Julio de 2011). Introducción a Google App Engine. Recuperado el Septiembre de 2014, de <http://www.genbetadev.com/programacion-en-la-nube/introduccion-a-google-app-engine>
- Colaboradores de Diario Linux. (2014). *Diario Linux*. Recuperado el 02 de Julio de 2014, de <http://diariolinux.com/2014/05/27/introduccion-a-open-edx-ii/>
- Colaboradores de Wikipedia. (5 de Mayo de 2014). *Miríada X*. Recuperado el 2 de Julio de 2014, de [http://es.wikipedia.org/w/index.php?title=Mir%C3%ADada\\_X&oldid=74217647](http://es.wikipedia.org/w/index.php?title=Mir%C3%ADada_X&oldid=74217647)
- Colaboradores de Wikipedia. (15 de Mayo de 2014). *Wikipedia - EDX*. Recuperado el 02 de Julio de 2014, de <http://es.wikipedia.org/w/index.php?title=EdX&oldid=74426662>
- Corso, C., & Alfaro, S. (2014). *Algoritmos de Data Mining aplicados en la enseñanza basada en la Web*. Córdoba: Universidad Tecnológica Nacional.
- Coursera. (2014). *Coursera*. Recuperado el 02 de Julio de 2014, de <https://www.coursera.org/about/>
- Coursera. (2014). <https://www.coursera.org>. Obtenido de <https://www.coursera.org>
- COURSESites Terms of use*. (1 de Septiembre de 2014). Recuperado el 23 de Septiembre de 2014, de COURSESites by Blackboard: [https://www.coursesites.com/webapps/Bb-sites-course-creation-BBLEARN/pages/CourseSitesTOU\\_Final.pdf](https://www.coursesites.com/webapps/Bb-sites-course-creation-BBLEARN/pages/CourseSitesTOU_Final.pdf)
- Creed-Dikeogu, G., & Clark, C. (01 de 01 de 2013). Are You MOOC-ing Yet? A Review for Academic Libraries. *Driving Towards New Frontiers*, 3, 9-13.



- Cuellar M., G. (2012). Data warehouse, aspectos técnicos, características, usos, beneficios, componentes, herramientas OLAP. Colombia. Recuperado el 2014 de Agosto de 20, de <http://fccea.unicauca.edu.co/old/datawarehouse.htm>
- De Giusti, A. (2001). *Algoritmos, datos y programas* (1era. ed.). Buenos Aires, Argentina: Pearson Education.
- Delgado Kloos, C. (1 de Marzo de 2014). Extendiendo Google Course Builder mediante Proyectos Realistas en un curso de Master. *VAEP-RITA*, 2(1). Recuperado el 12 de Septiembre de 2014
- Digital Learning. (17 de Marzo de 2012). ¿Qué hace un servidor web como Apache? Recuperado el 10 de Septiembre de 2014, de <http://www.digitallearning.es/blog/apache-servidor-web-configuracion-apache2-conf/>
- Dubet, F. (2011). Los alumnos, la escuela y la institución. *La experiencia sociológica*.
- Ecured. (4 de Abril de 2014). Pentaho Data Integration (Kettle). *Tino*. Recuperado el 31 de Agosto de 2014
- Edunext. (2014). *Edunext*. Recuperado el 2 de Julio de 2014, de <http://www.edunext.co/es/open-edx/>
- Edx. (2014). *www.edx.org*. Recuperado el 12 de 05 de 2014, de <https://www.edx.org/how-it-works>
- Eguiluz, J. (2013). *Introducción a XHTML*.
- Eguiluz Montero, J. (2013). Introducción a JavaScript.
- El Ahrache, S. I., Badir, H., Tabaa, Y., & Medouri, A. (05 de Mayo de 2013). Massive Open Online Courses : A New Dawn for Higher Education? *International Journal on Computer Science and Engineering (IJCSE)*, 5, 323-327.
- El Universo. (20 de 12 de 2011). Cerca de 60 mil bachilleres no irán a la Universidad: Senecyt. *El Universo*.
- Fernández, C. (2012). ¿Qué es un Data Warehouse? Recuperado el 4 de Septiembre de 2014, de <http://www.dataprix.com/qu-es-un-data-warehouse>
- Goicochea, A. (11 de Agosto de 2009). CRISP-DM. Recuperado el 12 de Septiembre de 2014, de <http://anibalgoicochea.com/2009/08/11/crisp-dm-una-metodologia-para-proyectos-de-mineria-de-datos/>
- Google. (2014). *Course Builder Wiki*. Recuperado el 02 de Julio de 2014, de <https://code.google.com/p/course-builder/wiki/CourseBuilderChecklist>
- Google. (2014). *Course Builder Wiki*. Recuperado el 2 de Julio de 2014, de <https://code.google.com/p/course-builder/wiki/CBRoles>
- Google. (6 de Junio de 2014). What is Google App Engine? Recuperado el 10 de Agosto de 2014, de <https://developers.google.com/appengine/docs/whatisgoogleappengine?hl=es>



- Goyal, M., & Vohra, R. (Marzo de 2012). Applications of Data Mining in Higher Education. *International Journal of Computer Science Issues*, 9(1), 113-120. Recuperado el 1 de Septiembre de 2014
- Gravitar. (2014). *Pentaho*. Recuperado el 31 de Agosto de 2014, de <http://gravitar.biz/pentaho/>
- Grünewald, F., Meinel, C., Totschnig, M., & Willems, C. (2013). Designing MOOCs for the Support of Multiple Learning Styles. *Lecture Notes in Computer Science*, 8095 2013, 371–382.
- Hafner, K. (16 de Abril de 2010). *An Open Mind*. Recuperado el 23 de Septiembre de 2014, de The New York Times: <http://www.nytimes.com/2010/04/18/education/edlife/18open-t.html>
- Hill, P. (9 de Noviembre de 2012). *Canvas Network – Are the LMS and MOOC Markets Colliding?* Recuperado el 23 de Septiembre de 2014, de e-Literate: <http://mfeldstein.com/canvas-network-are-the-lms-and-mooc-markets-colliding/>
- Hipp, J., Guntzer, U., & Nakhaeizadeh, G. (2000). Algorithms for Association Rule Mining: A General Survey and Comparison. *SIGKDD Explorations*.
- Holotescu, C., Cretu, V., & Grosseck, G. (2013). MOOC'S ANATOMY: MICROBLOGGING AS THE MOOC'S CONTROL CENTER. *Conference proceedings of "eLearning and Software for Education"*, 312-319.
- International Organization for Standarization. (2010). *System and software engineering - Systemsd and software quality requirements and Evaluation (SQuaRE) - System and software quality models*. G  neva, Suiza: ISO/IEC FDIS 25010.
- Khadijah Mohamad, S., & Zaidatun, T. (2013). *Educational data mining: A review*. Universidad Tecnol  gica de Malasia, Departamento de Ciencias de la educaci  n, matem  ticas y multimedia. Skudai: Procedia: Social and Behavioral Sciences. Recuperado el 30 de Agosto de 2014
- Khan Academy Fact pack*. (1 de Junio de 2014). Recuperado el 23 de Septiembre de 2014, de Khan Academy: <https://dl.dropboxusercontent.com/u/33330500/KAPressFactPack.pdf>
- Kop, R., & Hill, A. (2008). (The International Review of Research in Open and Distance Learning) Obtenido de <http://www.irrodl.org/index.php/irrodl/article/view/523/1137>
- Maldonado, J. (2013). Diapositivas realizadas para la materia de Algoritmos, Datos y Estructuras. Cuenca, Azuay, Ecuador.
- Marauri Mart  nez de Rituerto, P. M. (2014). FIGURA DE LOS FACILITADORES EN LOS CURSOS ONLINE MASIVOS Y ABIERTOS (COMA / MOOC): NUEVO ROL PROFESIONAL PARA LOS ENTORNOS EDUCATIVOS EN ABIERTO. *RIED*, 35-67.
- Marciales, A. (24 de enero de 2014). Ciclo de vida de la metodolog  a de Kimball. Recuperado el 12 de Septiembre de 2014, de <http://alvaromarciales.blogspot.com/2014/01/ciclo-de-vida-de-la-metodologia-de.html>
- Matas, D. (10 de Octubre de 2012). *Home Schooling Spain*. Recuperado el 02 de Julio de 2014, de <http://www.homeschoolingspain.com/2012/10/coursera-cursos-online-gratuitos.html?m=1>



- McAuley, A., Siemes, G., & Cormier, D. (2010). *The MOOC model for digital practice. SSHRC Knowledge Synthesis Grant on the Digital Economy*. Recuperado el 10 de 06 de 2014, de [http://www.edukwest.com/wp-content/uploads/2011/07/MOOC\\_Final](http://www.edukwest.com/wp-content/uploads/2011/07/MOOC_Final)
- Medina-Salguero, R., & Aguaded, I. (2014). LOS MOOC EN LA PLATAFORMA EDUCATIVA MIRIADAX. *Profesorado*, 18(1), 137-153.
- Méndez García, C. (2013). Diseño e implementación de cursos abiertos masivos en línea (MOOC): expectativas y consideraciones prácticas. *RED. Revista de Educación a Distancia*.
- Meza Escallón, A. (13 de Junio de 2014). *Arquitectura y Desarrollo de Software*. Recuperado el 02 de Julio de 2014, de <http://arquitecturadesarrollosoftware.blogspot.com/2013/06/como-colaborar-con-la-plataforma-mooc.html>
- Microsoft. (2 de Octubre de 2007). Escalabilidad. Recuperado el 2 de Septiembre de 2014, de [http://technet.microsoft.com/es-es/library/cc753794\(v=ws.10\).aspx](http://technet.microsoft.com/es-es/library/cc753794(v=ws.10).aspx)
- Microsoft. (18 de Marzo de 2010). ¿Qué es Excel? Recuperado el 10 de Septiembre de 2014, de <http://office.microsoft.com/es-mx/support/que-es-excel-HA010265948.aspx>
- Miriada X. (2014). *Universidades e instituciones*. Recuperado el 2 de Agosto de 2014, de <https://www.miriadax.net/>
- MSDN. (2014). *Algoritmos de minería de datos (Analysis Services: Minería de datos)*. Recuperado el 1 de Septiembre de 2014, de <http://msdn.microsoft.com/es-es/library/ms175595.aspx>
- MSDN. (2014). *Conceptos de minería de datos*. Recuperado el 1 de Septiembre de 2014, de <http://msdn.microsoft.com/es-es/library/ms174949.aspx>
- Norvig, P. (2012). *Helping the World to Teach*. Obtenido de Research Blog de Google: <http://googleresearch.blogspot.com/2012/09/helping-world-to-teach.html>
- Octeto. (28 de Noviembre de 2012). Recuperado el 10 de Agosto de 2014, de <http://cent.uji.es/octeto/node/3778>
- Organización JSON. (12 de Enero de 2011). Introducción a JSON. Recuperado el 9 de Septiembre de 2014, de <http://www.json.org/json-es.html>
- Peláez López, R. (s.f.). IMPACTO DEL APRENDIZAJE EN LOS ALUMNOS DE LAS UNIVERSIDADES DE GUAYAQUIL CON LA IMPLEMENTACIÓN DE UN MOOC UTILIZANDO PLATAFORMAS ONLINE.
- Pérez Costoya, F. (2012). *Tecnologías Google para procesamiento masivo de datos*.
- Pérez Valdés, D. (26 de Octubre de 2007). ¿Qué son las bases de datos? Recuperado el 10 de Septiembre de 2014, de <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>
- Pernías Peco, P., & Marco Such, M. (2007). Motivación y valor del proyecto OpenCourseWare: la universidad del siglo XXI. *Universidad y Sociedad del Conocimiento*, 48-57.



- Programa de doctorado Tecnologías Industriales. (2013). *Aplicaciones de la Inteligencia Artificial en Robótica*. Recuperado el 30 de Agosto de 2014
- Ramos, S. (2011). Procesos ETL. Bogotá, Colombia. Recuperado el 25 de Agosto de 2014
- rhous. (6 de Noviembre de 2011). Base de datos, tipos de lenguaje de programación y tipos de sistemas operativos. Recuperado el 31 de Agosto de 2014, de <http://rhous.wordpress.com/2011/11/06/base-de-datos-tipos-de-lenguaje-de-programacion-y-tipos-de-sistemas-operativos/>
- Rivadera, G. (2010). *La metodología de Kimball para el diseño de almacenes de datos (Data warehouse)*. Salta. Recuperado el 2 de Septiembre de 2014
- Rodríguez, A. S., & Concheiro, L. (10 de 9 de 2013). *La rotativa*. Recuperado el 15 de 06 de 2014, de <http://larotativa.nexos.com.mx/?p=115>
- Romo Uriarte, J., Benito Gómez, M., Portillo Berasaluce, J., & Casquero Oyarzabal, O. (30 de Junio de 2014). *OpenCourseWare: una alternativa para la publicación en contenidos educativos*. Obtenido de [http://www.researchgate.net/publication/220835761\\_OpenCourseWare\\_una\\_alternativa\\_para\\_la\\_publicacin\\_en\\_abierto\\_de\\_contenidos\\_educativos/file/72e7e520b566ca7542.pdf](http://www.researchgate.net/publication/220835761_OpenCourseWare_una_alternativa_para_la_publicacin_en_abierto_de_contenidos_educativos/file/72e7e520b566ca7542.pdf)
- Salcedo Parra, O., Galeano, R. M., & Rodriguez, L. (17 de Noviembre de 2009). *Metodología CRISP para la implementación de Data Warehouse*. Investigación , Bogotá. Recuperado el 7 de Septiembre de 2014
- Sánchez Cabañas, G. (Mayo de 2011). Extracción y análisis de información de Twitter con Google App Engine. Madrid, España.
- Sánchez, J. (2 de Abril de 2011). Programa Open Source WEKA. 146. Acarigua, Portuguesa, Venezuela: Universidad Nacional Abierta. Recuperado el Agosto de 30 de 2014
- Scardilli, B. (2013). Moocs: Classes for the Masses. *Information Today*, 32-34.
- Shaw, C. (20 de Diciembre de 2012). *FutureLearn is UK's chance to 'fight back', says OU vice-chancellor*. Recuperado el 23 de Septiembre de 2014, de The Guardian: <http://www.theguardian.com/higher-education-network/blog/2012/dec/20/futurelearn-uk-moocs-martin-bean>
- Siemens, G. (2013). Massive Open Online Courses: Innovation in Education? *Open Educational Resources: Innovation, Research and Practice*, 5.
- Sinnexus. (2013). Datawarehouse. La Coruña, España. Recuperado el 4 de Septiembre de 2014, de [http://www.sinnexus.com/business\\_intelligence/datawarehouse.aspx](http://www.sinnexus.com/business_intelligence/datawarehouse.aspx)
- Smith, B., & Eng, M. (2013). MOOCs: A Learning Journey. *Lecture Notes in Computer Science*, 8038, 244-255.
- Summan. (2013). *Pentaho BI Platform Server*. Bogotá. Recuperado el 30 de Agosto de 2014, de <http://www.summan.com/pentaho/pentaho-bi-platform-server>



- Three types of MOOC*. (27 de Marzo de 2013). Recuperado el 10 de Agosto de 2014, de Research Methods Module: <http://ditresearchmethods.wordpress.com/2013/03/27/three-types-of-mooc/>
- Toro, J., & Robles, M. (2013). *Una mirada a los MOOC y su implementación en el aula tradicional*.
- Tschofen, C., & Mackness, J. (2012). *The International Review of Research in Open and Distance Learning*. Obtenido de <http://www.irrodl.org/index.php/irrodl/article/view/1143/2086>
- Udacity. (2014). <https://www.udacity.com>. Obtenido de <https://www.udacity.com>
- Udemy. (2014). <https://www.udemy.com>. Obtenido de <https://www.udemy.com/about/#welcome>
- Universia. (27 de Julio de 2014). *César Alierta y Emilio Botín lanzan la plataforma de educación online MiríadaX, la más importante del mundo en español y portugués*. Recuperado el 10 de Agosto de 2014, de <http://noticias.universia.com.do/en-portada/noticia/2014/07/27/1101334/cesar-alierta-emilio-botin-lanzan-plataforma-educacion-online-miriadax-importante-mundo-espanol-portugues.html>
- Universidad Carlos III de Madrid. (Marzo de 2014). Guía metodológica para la planificación, diseño e impartición de MOOCs. Madrid, España.
- Universidad Politécnica de Madrid. (2014). *OpenCourseWare de la Universidad Politécnica de Madrid*. Recuperado el 5 de Julio de 2014, de <http://ocw.upm.es/bfque-es-ocw>
- Universidad Politécnica de Valencia. (2012). *Guía para la creación de asignaturas publicadas en el portal OCW*. Recuperado el 3 de Julio de 2014, de <http://www.upv.es/contenidos/DOCENRED/infoweb/docenred/info/guiaasignaturas.pdf>
- Universidad Tecnológica Nacional de Santa Fé. (2013). Data warehouse - Datamining. Bogotá, Colombia. Recuperado el 18 de Agosto de 2014
- Valverde, A. (28 de Agosto de 2012). *Más que PDF*. Recuperado el 02 de Julio de 2014, de <http://masquepdf.blogspot.com/2012/08/primer-aproximacion-coursera.html?m=1>
- W3C. (2014). *Guía Breve de CSSS*. Obtenido de <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>
- Welsh, D., & Dragusin, M. (2013). The New Generation of Massive Open Online Course (MOOCS) and Entrepreneurship Education. *Small Business Institute Journal*, 9, 51-65.
- Wikipedia. (2013). *Bases de datos multidimensionales*. Recuperado el 2 de Septiembre de 2014, de [http://es.wikipedia.org/wiki/Base\\_de\\_datos\\_multidimensional#Implementaci.C3.B3n](http://es.wikipedia.org/wiki/Base_de_datos_multidimensional#Implementaci.C3.B3n)
- Wikipedia. (Marzo de 2013). BigTable. Recuperado el 10 de Agosto de 2014, de <http://es.wikipedia.org/wiki/BigTable>
- Wikipedia. (Agosto de 2014). Estilo de aprendizaje. Recuperado el Septiembre de 2014, de [http://es.wikipedia.org/wiki/Estilo\\_de\\_aprendizaje](http://es.wikipedia.org/wiki/Estilo_de_aprendizaje)



Wikipedia. (Enero de 2014). Google App Engine. Recuperado el 10 de Agosto de 2014, de [http://es.wikipedia.org/wiki/Google\\_App\\_Engine](http://es.wikipedia.org/wiki/Google_App_Engine)

Wikipedia. (Junio de 2014). *Intergaz de programación de aplicaciones*. Recuperado el 10 de Agosto de 2014, de [http://es.wikipedia.org/wiki/Interfaz\\_de\\_programaci%C3%B3n\\_de\\_aplicaciones](http://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones)

Wikipedia. (2014). Python. Recuperado el 10 de Agosto de 2014, de <http://es.wikipedia.org/wiki/Python>

Wikipedia. (5 de Abril de 2014). WEKA (Aprendizaje Automático). Recuperado el 30 de Agosto de 2014, de [http://es.wikipedia.org/wiki/Weka\\_\(aprendizaje\\_autom%C3%A1tico\)](http://es.wikipedia.org/wiki/Weka_(aprendizaje_autom%C3%A1tico))

Wukman, A. (14 de Diciembre de 2012). *OpenMOOC Offers Open Source Class Creation Software*. Recuperado el 23 de Septiembre de 2014, de Online Courses: <http://www.onlinecollegecourses.com/2012/12/14/spanish-universities-offer-open-source-mooc-creation-software/>

Yuan, L., & Powell, S. (2013). MOOCs and Open Education: Implications for Higher Education.



# ANEXOS

## ANEXO 1: CONTENIDO Y ACTIVIDADES DEL MOOC

### Unidad 1: Introducción. Conceptos Básicos.

El objetivo de este capítulo es presentar algunas definiciones básicas que servirán de conceptos introductorios para las demás unidades

#### a. ¿Qué es programar?

Para comenzar vamos a definir algunas palabras claves que se usarán en este curso:

- **Informática:** Es la ciencia que estudia el análisis y resolución de problemas utilizando computadoras.  
La palabra ciencia hace referencia a la relación con una metodología fundamentada y racional para el estudio y resolución de los problemas. En este sentido, la informática se vincula con las matemáticas e ingeniería.
- **Computadora:** Una computadora es una máquina digital y sincrónica, con cierta capacidad de cálculo numérico y lógico, controlada por un programa almacenado y con posibilidad de comunicación con el mundo exterior.

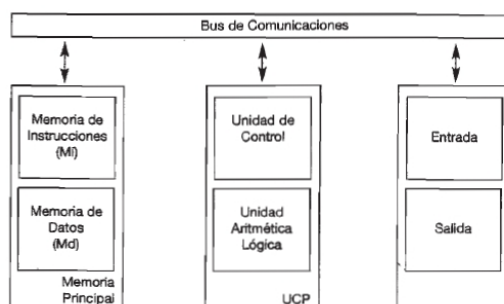


Figura A1.1: Estructura de una computadora.

Fuente: (De Giusti, 2001)

- **Programa:** Es un conjunto de instrucciones, ejecutables sobre una computadora, que permite cumplir una función específica.

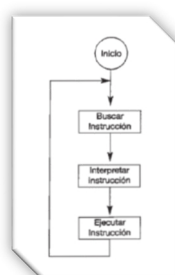


Figura A1.2: Estructura de un programa.

Fuente: (De Giusti, 2001)

En otras palabras, podemos decir que un programa nos permite representar, almacenar y transformar la información.



Figura A1.3: Distribución de un programa.

Fuente: Elaboración propia

- **Dato:** Es una representación simbólica (numérica, alfabética, algorítmica, etc.), un atributo o una característica de una entidad. El dato no tiene valor semántico (sentido) en sí mismo, pero si recibe un tratamiento (procesamiento) apropiado, se puede utilizar en la realización de cálculos o toma de decisiones.
- **Información:** Es un conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje.

### ACTIVIDADES

No.	PREGUNTA	TIPO	RESPUESTA
1	Una computadora es una máquina digital y asíncrona con capacidad de cálculo numérico y lógico.	Verdadero o Falso	Falso
2	Un programa nos permite representar, almacenar y transformar la información.	Verdadero o falso	Verdadero
3	Un dato es una representación simbólica, un atributo o una característica de una entidad.	Verdadero o falso	Verdadero

Tabla A1.1: Actividades de la lección 1 del módulo 1.

Fuente: Elaboración propia.

### MATERIAL ADICIONAL:

- **Video:** ¿Qué es programar? Disponible en: <http://youtu.be/Dtff3dArKoU>
- **Lecturas:**
  - **Lectura 1:** Conceptos Básicos. Pág. 4-8. Disponible en: <https://drive.google.com/open?id=0B2tOej64N8xJdkpIMjFINEp5WUk&authuser=0>
  - **Lectura 2:** Introducción a la Programación. Pág. 1-4. Disponible en: <https://drive.google.com/open?id=0B2tOej64N8xJY0gxUXgtUmplaEU&authuser=0>

#### b. ¿Qué es un lenguaje de programación?

Un lenguaje de programación es un lenguaje formal que han sido diseñados para expresar tareas de cómputo, es decir, para escribir programas.

Existen varios lenguajes de programación, entre ellos:

- Visual Basic
- Java
- C
- Smaltalk
- Pascal
- Python



Figura A1.4: Lenguajes de Programación.  
Fuente: (rhous, 2011)

## Python

Python es un lenguaje de programación interpretado o de script, que fue creado a principios de los años 90 por Guido Van Rossum, un científico de la computación e informático.

El nombre del lenguaje (Python) proviene de la afición de Van Rossum por los humoristas británicos Monty Python.

Python maneja una sintaxis limpia y favorece a un código legible en comparación con otros lenguajes de programación como Java o C++, ya que no hace uso de llaves ni de los puntos y comas al final de cada sentencia.

Se considera que Python es un lenguaje de programación multiparadigma ya que permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional.

Las principales características de Python son:

- Es interpretado, porque se ejecuta utilizando un lenguaje intermedio denominado intérprete, el cual va ejecutando instrucción por instrucción.
- Tipado dinámico, es decir, no se requiere declarar el tipo de dato de una determinada variable puesto que su tipo de dato se lo determinará en tiempo de ejecución según el tipo de valor asignado.
- Es fuertemente tipado, ya que no se permite realizar operaciones con variables de distintos tipos, es decir, si tenemos un cadena de caracteres no le podemos sumar un número entero.
- Es multiplataforma, puesto que el intérprete de Python está disponible para varios sistemas operativos como Windows, Linux, Mac, etc.

Las ventajas que se tiene al utilizar Python son:

- Fácil desarrollo de aplicaciones
- Es sencillo y rápido.
- Posee un entorno interactivo.
- Sus bibliotecas hacen gran parte del trabajo
- Es rico en estructuras de datos.
- Es gratuito



## ACTIVIDADES

No.	PREGUNTA	TIPO	RESPUESTA
1	Un lenguaje de programación es un lenguaje formal que han sido diseñados para expresar tareas de cómputo, es decir, para escribir programas.	Verdadero o Falso	Verdadero
2	Python es un lenguaje multiplataforma, es decir, puede ser utilizado en Windows, Linux, Mac, etc.	Verdadero o falso	Verdadero
3	Para definir una variable en Python se debe definir primero el tipo de dato que se almacenará.	Verdadero o falso	Falso

Tabla A1.2: Actividades de la lección 2 del módulo 1.

Fuente: Elaboración propia

## MATERIAL ADICIONAL:

- **Video:** Introducción a Python. Disponible en: <http://youtu.be/SY472Hf9DHc>
- **Lecturas:**
  - **Lectura 2** Introducción a la Programación. Pág. 4-5. Disponible en: <https://drive.google.com/open?id=0B2tOej64N8xJY0gxUXgtUmplaEU&authuser=0>
  - **Lectura 3** Introducción a Python. Disponible en: <https://drive.google.com/open?id=0B2tOej64N8xJQnFJRXdDN29QSWs&authuser=0>

### c. Modelización de programas del mundo real

El mundo real es naturalmente complejo, y en muchas ocasiones los problemas a resolver resultan difíciles de sintetizar.

Por ejemplo, si una persona desea viajar de Cuenca a Loja y dispone únicamente de \$12 para el mismo debe tomar decisiones:

- Si gasta su dinero en el pasaje de bus, no tendrá para la alimentación
- Si “jala dedo”, podrá comer el almuerzo y la merienda.

A raíz de esto, surgen dos definiciones importantes:

- **Abstracción:** Proceso de análisis del mundo real para interpretar los aspectos esenciales de un problema y expresarlo en términos precisos.
- **Modelización:** Abstraer un proceso del mundo real y simplificar su expresión, tratando de encontrar los aspectos principales que se pueden resolver (requerimientos), los datos que se han de procesar y el contexto del problema.

Ahora, consideremos el siguiente ejemplo. Pedimos a otra persona que piense un número del uno al diez. A otra persona le pedimos que adivine dicho número. Si lo hace, recibe un premio caso contrario, volverá a intentarlo.

El ejemplo consta de dos partes:

1. Pensar un número del 1 al 10.
2. Si adivina el número recibe el premio, caso contrario volverá a intentarlo.

El literal 1, hace referencia a una precondición, que es una información que se conoce como verdadera antes de iniciar un programa.

El literal 2, hace referencia a una poscondición, que es una información que debiera ser verdadera para concluir el programa, si se cumple adecuadamente el requerimiento pedido.

### La especificación de los problemas del mundo real

El proceso de analizar los problemas del mundo real y determinar en forma clara y concreta el objetivo que se desea se denomina especificación.

Especificar un problema significa establecer en forma unívoca el contexto, las precondiciones y el resultado esperado, de lo cual se derivan las poscondiciones.

La especificación es fundamental para convertir el problema real en un modelo que permita la utilización de computadoras en su solución. Para esto la escritura de un programa constituye el eje fundamental de la ciencia de la computación.

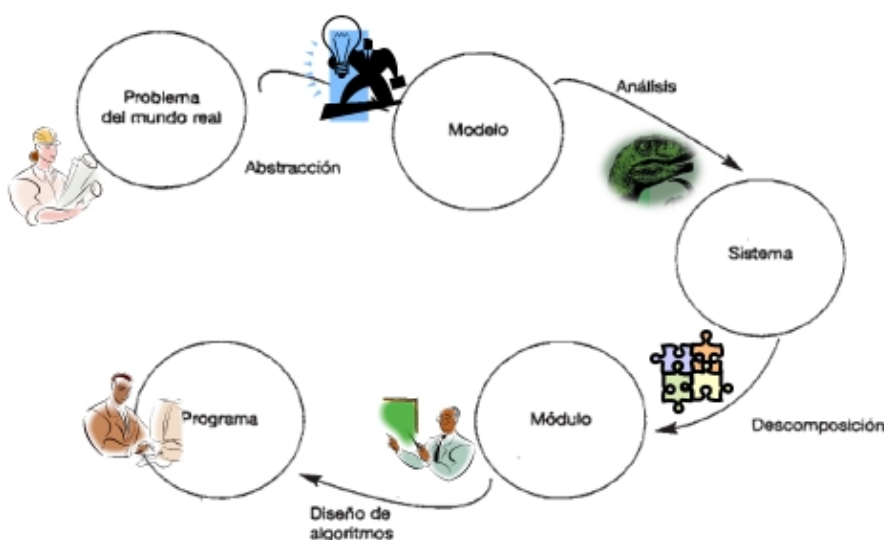


Figura A1.5: Pasos para realizar un programa.  
Fuente: (De Giusti, 2001)

### ACTIVIDADES

No.	PREGUNTA	TIPO	RESPUESTA
1	Para llegar a un modelo a partir de un problema se debe realizar un proceso de abstracción	Verdadero o Falso	Verdadero
2	A partir del modelo se puede elaborar el análisis de la solución como sistema	Verdadero o falso	Verdadero
3	Especificar un problema significa establecer en forma unívoca su contexto, precondiciones y el resultado esperado.	Verdadero o falso	Verdadero

Tabla A1.3: Actividades de la lección 3 del módulo 1.  
Fuente:

### MATERIAL ADICIONAL:

- **Video:** Modelización de problemas del mundo real. Disponible en: <http://youtu.be/L6WaD6s2kfU>
- **Lecturas:**



- **Lectura 1** Conceptos básicos. Pág. 7-13. Disponible en:  
<https://drive.google.com/open?id=0B2tOej64N8xJdkpIMjFINEp5WUk&authuser=0>

#### **d. Del problema real a su solución por computadora**

##### **Importancia del Contexto**

El contexto del problema real es muy importante para analizar y diseñar la solución utilizando computadoras.

En muchas ocasiones, el contexto impone restricciones que son importantes para la elaboración de la solución. Por ejemplo, si se está estudiando el software de un cajero automático, tener la respuesta a una consulta de saldo en menos de 10 minutos es un dato de contexto determinante sobre el tipo de solución aceptable.

La caracterización del contexto es muy amplia. Por ejemplo, el número de empleados que pueden ser atendidos simultáneamente, los clientes con diferentes prioridades, la posibilidad de pagos con tarjeta de crédito, el tipo y la cantidad de líneas telefónicas disponibles, etc., son datos que acompañan al modelo esencial del sistema y deben tenerse en cuenta antes de diseñar una solución.

##### **Descomposición en módulos**

La Modularización o descomposición del modelo es muy importante para la elaboración de la solución. Consiste en dividir al problema en problemas más pequeños, donde la fusión de los mismos, nos de la solución al problema

##### **Partes de un programa**

Un programa consta de dos componentes básicos:

- Instrucciones: Conocidas también como acciones, representan las operaciones que ejecutará la computadora al interpretar el programa.
- Datos: Son valores de información de los que se necesita disponer, y en ocasiones transformar, para ejecutar la función del programa. Los datos pueden ser de dos tipos: constantes o variables, los primeros son aquellos que no cambian durante la ejecución del programa, mientras que los últimos pueden variar durante la ejecución.

Tanto los datos constantes como los variables deben guardarse en la memoria de datos de una computadora. En ambos casos estarán representados simbólicamente por un nombre que se asocia a una dirección única de memoria. Por esto, el contenido de la dirección de memoria correspondiente a un dato constante se asigna una sola vez en los programas; en cambio, el contenido (valor) de la dirección de memoria correspondiente a un dato variable puede asignarse y cambiar muchas veces durante la ejecución del programa.

Además de datos e instrucciones, al leer un programa encontraremos comentarios, es decir, texto aclaratorio para el programador o el usuario, que no es entendido ni ejecutado por la computadora. Este texto sirve para clarificar qué hace un programa, y será de gran importancia cuando se intente modificarlo o corregirlo.

Una visión interesante de un programa es que corresponde a una transformación de datos. A partir de un contexto determinado por las precondiciones, el programa transforma la información y debería llegar al resultado esperado produciendo un nuevo contexto, caracterizado por las poscondiciones.

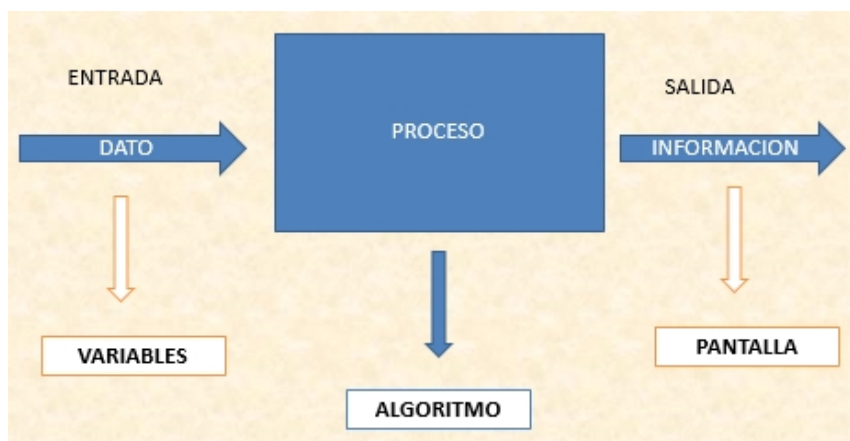


Figura A1.6: Visión general de un programa.

Fuente: (Maldonado, 2013)

## Algoritmo

Un algoritmo es la especificación rigurosa de la secuencia de pasos (instrucciones) a realizar sobre un autómata para alcanzar un resultado deseado en un tiempo finito.

El término “especificación rigurosa” hace referencia a que un algoritmo debe ser expresado de forma unívoca, es decir, que la sintaxis y la semántica de cada instrucción deben estar perfectamente definidas.

Un ejemplo de algoritmo lo podemos ver en la siguiente figura.

```

Algoritmo      Preparar una taza de té.
Entrada: tetera, taza, bolsa de té
Salida: taza de té

Inicio
  Tomar la tetera
  Llenarla de agua
  Encender el fuego
  Poner la tetera en el fuego
  Esperar a que hierva el agua
  Tomar la bolsa de té
  Introducirla en la tetera
  Esperar 1 minuto
  Echar el té en la taza
Fin
  
```

Figura A1.7: Ejemplo de un algoritmo.

Fuente: (De Giusti, 2001)

## ACTIVIDADES

No.	PREGUNTA	TIPO	RESPUESTA
1	El contexto del problema real es muy importante para analizar y diseñar la solución utilizando computadoras.	Verdadero o Falso	Verdadero
2	Para encontrar la solución a un problema, no es conveniente realizar la descomposición en módulos	Verdadero o falso	Falso





3	Los componentes básicos de un programa son el contexto y la modularización	Verdadero o falso	Falso
4	Un algoritmo es la especificación rigurosa de la secuencia de pasos (instrucciones) a realizar sobre un autómata para alcanzar un resultado deseado en un tiempo finito.	Verdadero o falso	Verdadero

Tabla A1.4: Actividades de la lección 4 del módulo 1.

Fuente: Elaboración propia.

## MATERIALES ADICIONALES

- **Video:** Del problema a la solución por computadora. Disponible en: <http://youtu.be/TfZXyr4ptG0>
- **Lecturas:**
  - **Lectura 1:** Conceptos básicos. Pág. 15-17. Disponible en: <https://drive.google.com/open?id=0B2tOej64N8xJdkpIMjFINEp5WUk&authuser=0>

### e. Conceptos sobre software

Se conoce como software al equipamiento lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware. (Wikipedia, 2014)

Los componentes lógicos incluyen, entre muchos otros, las aplicaciones informáticas; tales como el procesador de texto, que permite al usuario realizar todas las tareas concernientes a la edición de textos; el llamado software de sistema, tal como el sistema operativo, que básicamente permite al resto de los programas funcionar adecuadamente, facilitando también la interacción entre los componentes físicos y el resto de las aplicaciones, y proporcionando una interfaz de usuario.

### Etapas de la resolución de un problema con computadora

1. **Análisis del problema:** Es la primera etapa, en donde se analiza el problema en su contexto del mundo real. Deben obtenerse los requerimientos del usuario. El resultado de este análisis es un modelo preciso del ambiente del problema y del objetivo a resolver. Un componente importante de este modelo son los datos a utilizar y las transformaciones de los mismos que llevan al objetivo.
2. **Diseño de una solución:** El primer paso en el diseño de la solución es la modularización del problema, es decir, la descomposición del mismo en partes que tendrán una función bien definida y datos propios. A su vez, debe establecerse la comunicación entre los módulos del sistema de software propuesto.
3. **Especificación de algoritmos:** La elección del algoritmo adecuado para la función del módulo es muy importante para la eficiencia posterior del sistema de software. Una misma función sobre los mismos datos puede resolverse con una utilización de recursos muy diferente (memoria, tiempo) según el algoritmo elegido.
4. **Escritura de programas:** Un algoritmo es una especificación simbólica que debe convertirse en un programa real sobre un lenguaje de programación concreto. Este proceso de programación tiende a automatizarse en la medida que los lenguajes algorítmicos se acercan a los lenguajes reales de programación. A su vez, el programa escrito en un lenguaje de programación determinado debe traducirse (automáticamente) al lenguaje de máquina de la computadora que lo vaya a ejecutar.

Esta traducción, denominada compilación, permite detectar y corregir los errores sintácticos que se cometen en la escritura del programa.

5. **Verificación:** Una vez que se tiene un programa escrito en un lenguaje real y depurado de errores sintácticos, se debe verificar que su ejecución conduzca al resultado deseado, con datos representativos del problema real.

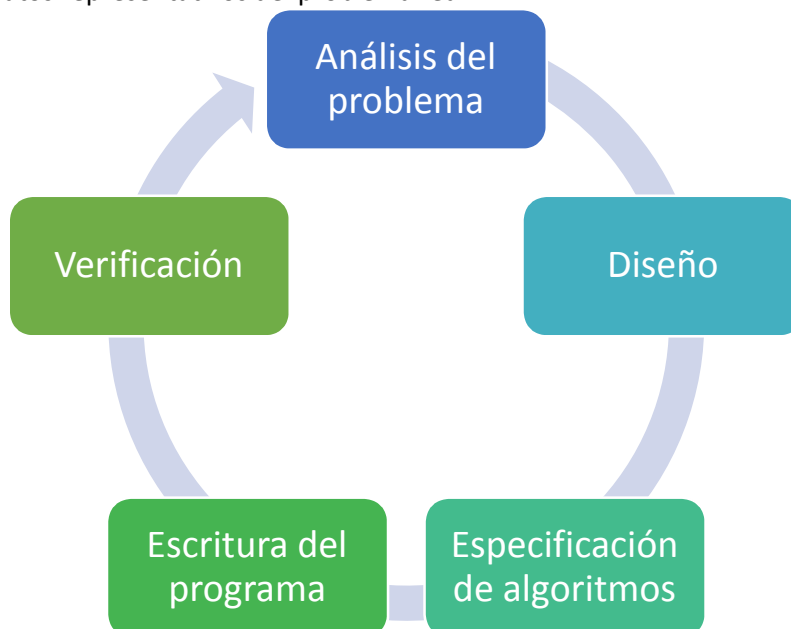


Figura A1.8: Etapas de resolución de un problema.  
Fuente: Elaboración propia

## ACTIVIDADES

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	Se conoce como software al equipamiento lógico o soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware.	Verdadero o Falso		Verdadero
2	Ordene: Las etapas en la resolución de problemas por computadora son: 1. Análisis del problema 2. Especificación de algoritmos 3. Verificación 4. Diseño de una solución 5. Escritura de Programas	Opción múltiple con una respuesta	a) 1,2,3,4,5 b) 1,4,2,5,3 c) 1,2,4,3,5 d) 4,1,3,2,5	b

Tabla A1.5: Actividades de la lección 5 del módulo 1.  
Fuente: Elaboración propia.

## MATERIALES ADICIONALES

- **Video:** Conceptos sobre software. Disponible en: <http://youtu.be/2i9R4e8n9o8>
- **Lecturas:**
  - **Lectura 1:** Conceptos básicos. Pág. 17-19. Disponible en: <https://drive.google.com/open?id=0B2tOej64N8xJdkplMjFINEp5WUK&authuser=0>



## Unidad 2: Algoritmos

### a. Concepto de algoritmo

Un algoritmo es un conjunto de instrucciones que nos sirven para ejecutar una tarea o resolver un problema, el desarrollo de algoritmos es parte fundamental en el desarrollo de un programa y debe cumplir ciertas características como ser legible, modular y estructurado.

En otras palabras, podemos definir a los algoritmos, como una secuencia finita de pasos que permiten resolver un problema.

La palabra algoritmo deriva del nombre del matemático árabe Al Juarismi.

Los algoritmos deben cumplir con las siguientes características:

- **Entrada:** definir lo que necesita el algoritmo
- **Salida:** definir el resultado.
- **Finito:** el algoritmo termina en un número finito de pasos.
- **Correcto:** hace lo que se supone que debe hacer. La solución es correcta
- **Efectividad:** cada instrucción se completa en tiempo finito.
- **General:** debe ser lo suficientemente general como para contemplar todos los casos de entrada.

Un ejemplo de un algoritmo para cambiar la llanta del carro lo podemos observar a continuación:

**Entrada:** Llanta de repuesto, gata, llave de tuerca

**Salida:** Llanta cambiada

**Proceso:**

Inicio

Ubicar la gata debajo del carro

Accionar la gata

Tomar la llave de tuercas

Colocar la llave de tuercas sobre la tuerca 1

Girar hasta aflojarla y retirar

Colocar la llave de tuercas sobre la tuerca 2

Girar hasta aflojarla y retirar

Retirar la llanta dañada

Tomar la llanta de repuesto y ubicarla en el lugar

Tomar la tuerca 1 y colocarla

Tomar la llave de tuercas y ajustarla

Tomar la tuerca 2 y colocarla

Tomar la llave de tuercas y ajustarla



Bajar y sacar la gata

Llanta cambiada

Fin

## ACTIVIDADES

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	Un algoritmo es una secuencia finita de pasos que permiten resolver un problema.	Verdadero o Falso		Verdadero
2	Las características de un algoritmo son:	Opción múltiple con varias respuestas	a) Infinito, Definido y Efectividad b) Entrada, Salida, Finito, Correcto, Efectividad y General. c) Sin solución, Finito, Ineficiente, General d) Eficiente, Infinito, Sin solución e) Finito e Ineficiente	b

Tabla A1.6: Actividades de la lección 1 del módulo 2.

.Fuente: Elaboración propia.

## MATERIAL ADICIONAL:

- **Video:** ¿Qué es un algoritmo? Disponible en: <http://youtu.be/Mu3K-8RpvEU>

### b. Concepto de variable

Una **Variable** es una estructura de datos, que puede cambiar su valor según la situación deseada.

Para crear una variable se debe tener algunas consideraciones al momento de asignarles un nombre:

- Puedes usar números y letras pero debe comenzar con una letra.
- Puedes usar mayúsculas y minúsculas, por convenio se suelen usar las minúsculas.
- También es válido el guion bajo “\_”.
- Totalmente prohibido el uso de tildes en los nombres de variable

### Representación de una variable

Toda variable ocupará un espacio en memoria, este espacio variará según el tipo de dato que este represente, a continuación unos ejemplos:

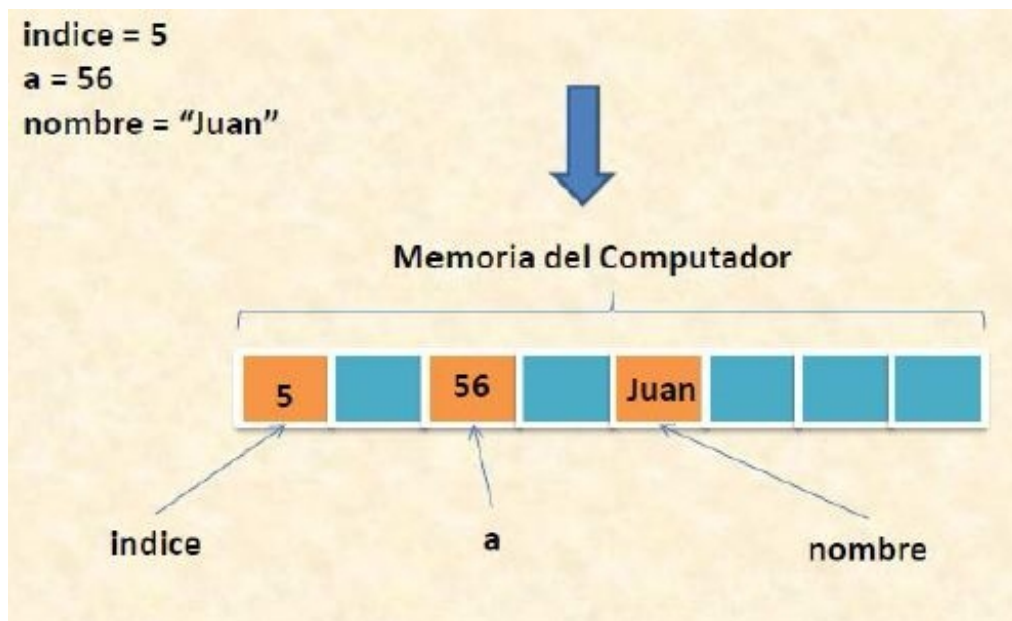


Figura A1.9: Ejemplo 1 de la representación de una variable.

Fuente: (Maldonado, 2013)

Para poder acceder a estos espacios de memoria se necesita de los nombres de las variables, a continuación se presenta un ejemplo en el que se leen y escriben valores en distintas variables para poder calcular la pendiente de una recta:

- **Formula:** 
$$m = \frac{y_2 - y_1}{x_2 - x_1}$$
- **Entrada:** x1,y1,x2,y2
- **Salida:** m
- **Proceso:**

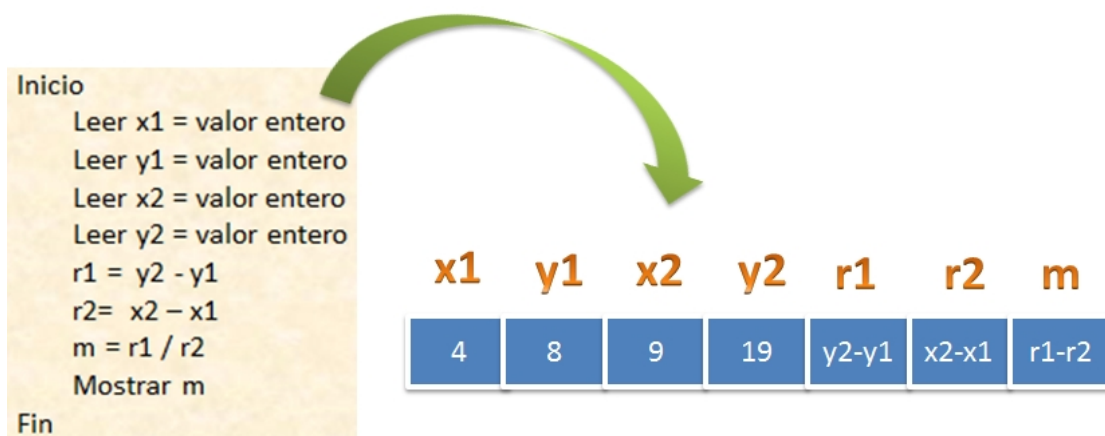


Figura A1.10: Ejemplo 2 de la representación de una variable.

Fuente: (Maldonado, 2013)

### Del Algoritmo al Lenguaje de Programación

El lenguaje de programación que será manejado a lo largo de este curso es Python, a continuación se detallan dos de las operaciones fundamentales:

- **Lectura de datos por teclado:** se emplea la sentencia

**input("mensaje")**

En donde "mensaje" representa un texto que aparecerá en pantalla junto a la parte en donde se tomarán los datos tomados ingresados por teclado.

- **Salida de datos por pantalla:** se emplea la sentencia

**print(nombre de variable)**

En donde nombre de variable indicará la variable de la cual se tomará el valor para mostrar en pantalla.

**Ejemplo:** a continuación se muestra el paso del algoritmo empleado en el ejemplo anterior hacia la sintaxis de Python.

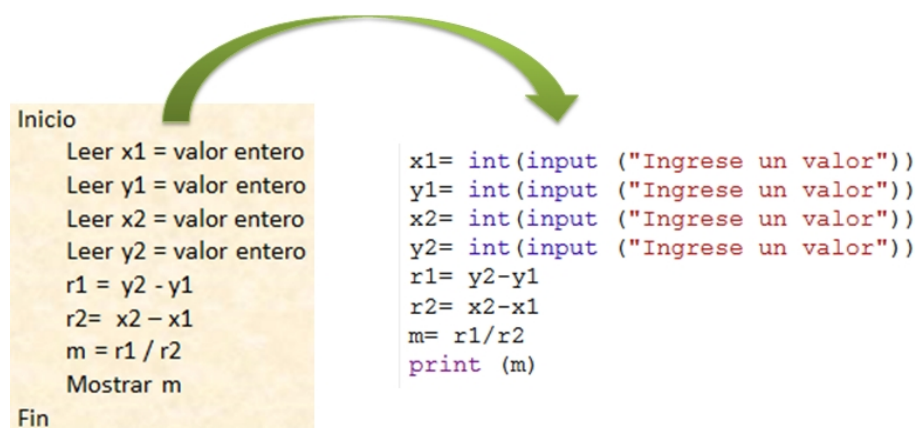


Figura A1.11: Ejemplo de sintaxis en Python.  
Fuente: (Maldonado, 2013)

## ACTIVIDADES

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	Una variable es una estructura que no puede cambiar su valor a lo largo de la ejecución del programa.	Verdadero o Falso		Falso
2	Una variable ocupa un espacio en memoria.	Verdadero o Falso		Verdadero
3	¿Cuál de estas es o cuales son nombres de variables correctas?	Opción múltiple con varias respuestas	a) num1, valor, variable_1 b) número, 1valor, variable_1 c) 1valor, número	a
4	La sentencia input permite:	Opción múltiple con una respuesta	a) salida de datos en pantalla b) lectura de datos por teclado	b

			c) a y b son correctas d) ninguna de las anteriores	
--	--	--	--	--

Tabla A1.7: Actividades de la lección 2 del módulo 2.

Fuente: Elaboración propia.

**MATERIAL ADICIONAL**

- **Video:** 02 Curso de Programación en Python: Variables, E/S, Operaciones Básicas. Disponible en: <http://youtu.be/dk79gnaUTfw>
- **Lectura:**
  - **Lectura 2 Tipos de datos simples;** páginas: 9-11; disponible en: <https://drive.google.com/file/d/0B2tOej64N8xJNU81X2JtbzL4VEE/edit?usp=sharing>
  - **Lectura 4 variables, sentencias, expresiones;** páginas: 2-9; disponible en: <https://drive.google.com/file/d/0B2tOej64N8xJYnRpZTIEYWhsSTg/edit?usp=sharing>
  - **Lectura 5 Tipos de datos;** páginas: 1-3; disponible en: <https://drive.google.com/file/d/0B2tOej64N8xJUkxiRkhrNkZjckE/edit?usp=sharing>

**c. Tipos de datos que almacenan las variables**

Un tipo de dato es una clase de objetos ligados a un conjunto de operaciones para crearlos y manipularlos.

Al definir un tipo de dato lo que se está indicando es la clase de valores que pueden tomar sus elementos y las operaciones que se pueden realizar sobre ellos.

Los tipos de datos se caracterizan por:

- Un rango de valores posibles.
- Un conjunto de operaciones realizables sobre ese tipo
- Su representación interna

Los tipos de datos simples o primitivos son:

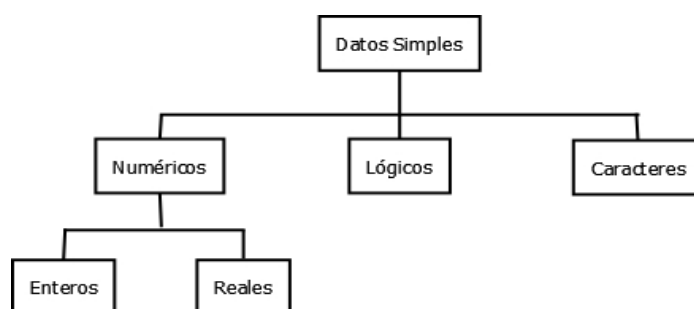


Figura A1.12: Tipos de datos primitivos.

Fuente: (De Giusti, 2001)

- **Numéricos:** Conjunto de valores numéricos que pueden representarse de dos formas:
  - **Enteros:** Es el tipo de datos numéricos más simple. Los elementos de este tipo son: ..., -3, -2, -1, 0, 1, 2, 3, ...
  - **Reales:** Son una clase de datos numéricos que permiten representar números decimales. Los valores fraccionarios forman una serie ordenada, desde un valor mínimo negativo, hasta un valor máximo determinado por la norma IEEE 754, de 1985, pero los valores no están distribuidos de manera uniforme en ese



intervalo, como sucede con los enteros. Algunos ejemplo de dato reales son: -43,222 9,5748 0,45

Los tipos de datos numéricos cuentan con un conjunto de operaciones posibles que son:

- Suma (+)
- Resta (-)
- Multiplicación (\*)
- División (/)
- Módulo (Mod): Permite obtener el resto de una división

Los símbolos que se encuentran entre paréntesis son conocidos como operadores aritméticos.

Si tenemos la expresión 5+6, los valores 5 y 6 se denominan operandos y al valor de la expresión se lo conoce como resultado.

- **Lógicos:** Conocidos también como booleanos, en honor al matemático británico George Boole, quien desarrolló el álgebra lógica o de Boole. Es un dato que puede tomar un valor entre un conjunto formado por dos posibles. Dichos valores son:
    - Verdadero (true)
    - Falso (false)
  - **Carácter:** Proporciona objetos de la clase de datos que contienen solo un elemento como su valor. Este conjunto de elementos está establecido y normalizado por un estándar llamado ASCII (American Code for Information Interchange), el cual establece cuales son los elemento y el orden de precedencia entre los mismos. Los elementos son las letras, números y símbolos especiales disponibles en el teclado de la computadora y algunos otros elementos gráficos. Cabe recalcar que el código ASCII no fue único, pero si el más utilizado.
    - Letras minúsculas: 'a' 'b' 'c'...'y' 'z'
    - Letras mayúsculas: 'A' 'B' 'C'...'Y' 'Z'
    - Dígitos: '0' '1' '2'...'8' '9'
    - Caracteres especiales: '!' '#' '\$' '+'
- Se debe tener en cuenta que no es lo mismo el valor entero 0 que el símbolo carácter '0'.

## ACTIVIDADES

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	Los tipos de datos numéricos son conocidos también como booleanos.	Verdadero o Falso		Falso
2	Los tipos de datos se clasifican en:	Opción múltiple con varias respuestas	a) Reales, Enteros b) Carácter, Enteros, Booleanos c) Booleanos, Carácter, Numéricos	c

Tabla A1.8: Actividades de la lección 3 del módulo 2.

Fuente: Elaboración propia

## MATERIAL ADICIONAL

- **Video:** Tipos de datos. Disponible en: <http://youtu.be/ug9kbFGt8yI>

- **Lectura:**

- **Lectura 2 Tipos de datos simples;** páginas: 2-9; disponible en: <https://drive.google.com/file/d/0B2tOej64N8xJNU81X2JtbzI4VEE/edit?usp=sharing>
- **Lectura 5 Tipos de datos;** páginas: 4-12; disponible en: <https://drive.google.com/file/d/0B2tOej64N8xJUkxiRkhrNkZjckE/edit?usp=sharing>

- d. **Representación de un algoritmo con Diagramas de flujo**

Un diagrama de flujo de datos es una descripción gráfica de un algoritmo para la resolución de problemas.

- Los diagramas de flujo de datos están conformados por figuras conectadas con flechas.
- Para ejecutar un proceso descrito por un diagrama de flujo de datos se comienza por el INICIO y se siguen las flechas de figura a figura, ejecutándose las acciones indicadas por cada figura; el tipo de figura indica el tipo de paso que representa.
- Dicho de otra forma es otra manera de definir un algoritmo.

### Simbología de los Diagramas de Flujo

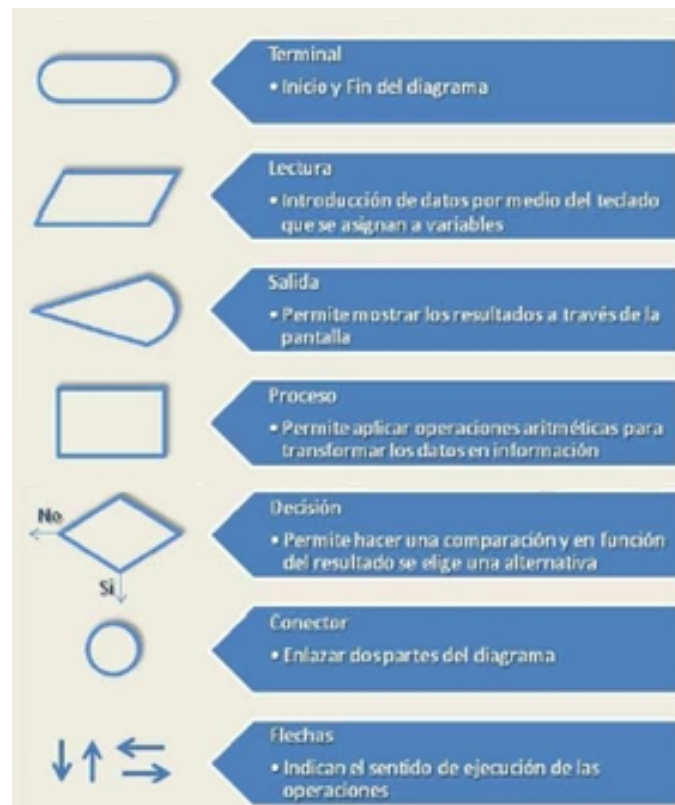


Figura A1.13: Simbología de los diagramas de flujo.

Fuente: (Maldonado, 2013)

**Ejemplo:** algoritmo para el cálculo de la pendiente de una recta.

- **Entrada:**  $x_1$ ,  $y_1$ ,  $x_2$ ,  $y_2$
- **Salida:**  $m$
- **Proceso:**

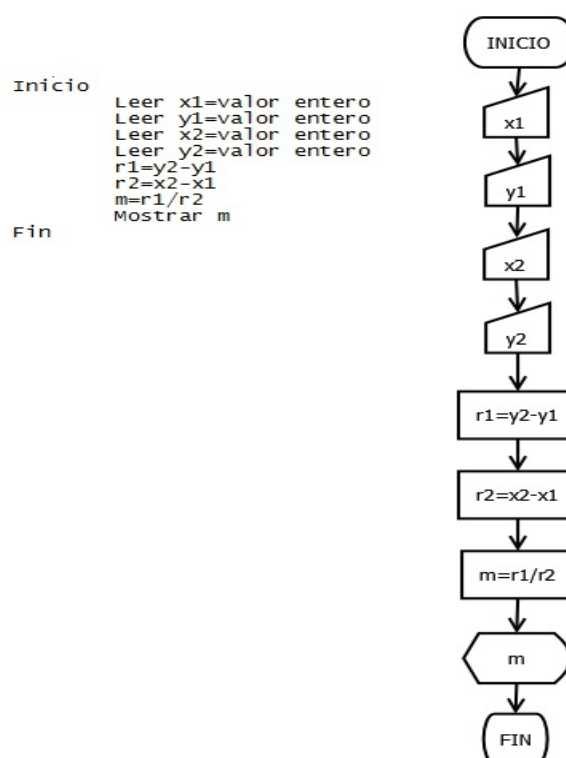


Figura A1.14: Ejemplo de algoritmo con diagramas de flujo.  
Fuente: (De Giusti, 2001)

## ACTIVIDADES

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	Los diagramas de flujo no permiten representar un algoritmo.	Verdadero o Falso		Falso
2	Los diagramas de flujo están conformados por figuras conectadas por flechas.	Verdadero o Falso		Verdadero
3	Un diagrama de flujo está compuesto por:	Opción múltiple con una respuesta	a) Flechas b) Figuras c) Ambas d) Ninguna de las anteriores	c
4	En un diagrama de flujo la forma de la figura indicará el tipo de paso que se realizará.	Verdadero o Falso		Verdadero

Tabla A1.9: Actividades de la lección 4 del módulo 2.  
Fuente: Elaboración propia.

## MATERIAL ADICIONAL

- **Video:** Diagramas de Flujo. Disponible en: <http://youtu.be/udN-1T4FWPA>
- **Lectura:**
  - **Aprenda a crear Diagramas de flujo**, disponible en: <http://mis-algoritmos.com/aprenda-a-crear-diagramas-de-flujo>

### e. Representación de algoritmos con Pseudocódigo

Es una forma de representación de algoritmos en la que se utiliza expresiones semejantes a las de los lenguajes de programación pero expuestas de tal manera que faciliten la lectura humana.

#### Estándar para Pseudocódigo

##### Inicio de programa:

INICIO <nombre del programa>

##### Fin de programa

FIN <nombre del programa>

##### Declaración de variables:

Tipo <nombre variable 1>, <nombre variable 2>, ..., <nombre variable n>

##### Asignación:

<nombre variable> = <expresión>

##### Lectura desde pantalla:

Leer <nombre de variable>

Si se desea provocar un salto de línea luego de la lectura se puede usar:

Leer y saltar <nombre de variable>

##### Escritura a pantalla:

Escribir <nombre de variable>

Escribir "<texto>"

Es posible concatenar información para escribir en pantalla usando una o varias veces el signo más +. Por ejemplo:

Escribir "<texto>" + <nombre de variable>

Escribir "<texto>" + <nombre de variable> + "<texto>"

Si se desea provocar un salto de línea luego de la escritura se puede usar:

Escribir y saltar "<texto>"

Figura A1.15: Estándar de pseudocódigo.

Fuente: (De Giusti, 2001)

**Ejemplo:** algoritmo para el cálculo del perímetro de un triángulo rectángulo.

- **Fórmula:**  $p=a+b+c$
- **Entrada:** a,b,c
- **Salida:** p
- **Proceso:**

```

INICIO
    Leer a
    Leer b
    Leer c
    p=a+b+c
    mostrar p
FIN

```

Figura A1.16: Ejemplo de pseudocódigo.  
Fuente: Elaboración propia.

## ACTIVIDADES

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	El pseudocódigo intenta asemejar la sintaxis empleada por los lenguajes de programación a una sintaxis que facilite la lectura humana.	Verdadero o Falso		Verdadero
2	¿Cuáles de estas palabras pertenecen al pseudocódigo?	Opción múltiple con una respuesta	a) Leer b) Escribir c) INICIO d) a y b e) b y c f) a, b y c	f

Tabla A1.10: Actividades de la lección 5 del módulo 2.  
Fuente: Elaboración propia

## MATERIAL ADICIONAL

- **Video:** Pseudocodigo. Disponible en: [http://youtu.be/6Prol\\_23sXA](http://youtu.be/6Prol_23sXA)
- **Lectura:**
  - **Estándar Pseudocodigo;** página: 1; disponible en: <https://drive.google.com/file/d/0B2tOej64N8xJdDICT2ZOMIN1SGM/edit?usp=sharing>

### f. Importancia de documentar un algoritmo

Un programa bien documentado será más fácil de leer y mantener. Para ello la documentación es fundamental. Es común que la mayoría de los lenguajes de programación provean algún mecanismo de documentación, por ejemplo, a través de la inserción de comentario en el programa. Un programa sin comentarios revela un estilo de programación pobre y peligroso, ya que dificulta el mantenimiento adecuado del mismo.

Es recomendable utilizar un comentario general para el objetivo del programa o del módulo de programa en cuestión, que refleje la especificación del problema a resolver, de la forma más completa posible. Su lectura debe ser suficiente para poder entender las acciones que se llevan a cabo en el mismo. Además, deberán describirse todos los datos variables y constantes que intervengan en el programa, e indicar la fecha de realización del mismo, el autor, etc.

Por otra parte, se deberán elegir identificadores de manera tal que sean autoexplicativos. En cuanto a los comentarios intercalados en el programa (documentación on-line o interna), deben realizarse con cierto criterio, pues deben contribuir a la claridad del programa; por ejemplo, es importante comentar el estado inicial y final entre las acciones de una determinada secuencia, los efectos colaterales probables que puedan tener lugar luego de una llamada a un módulo, etc.



Es importante destacar que cuando se realiza el mantenimiento de un programa no solo se actualiza el código, sino también los comentarios del mismo. Debe quedar claro que los comentarios no son un agregado al programa, sino una parte importante del mismo.

### ACTIVIDADES

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	Un programa bien documentado será más fácil de leer y mantener.	Verdadero o Falso		Verdadero
2	Cuando se realice el mantenimiento de un programa se debe actualizar:	Opción múltiple con varias respuestas	a) Código, Algoritmo b) Algoritmo, Variables c) Código, Comentarios d) Variables, Algoritmos	c

Tabla A1.11: Actividades de la lección 6 del módulo 2.

Fuente: Elaboración propia

### MATERIAL ADICIONAL

- **Video:** Importancia de documentar un algoritmo. Disponible en: <http://youtu.be/HO9zbHxW8xM>
- **Lectura:**
  - **Lectura 1 La Corrección de algoritmos;** Disponible en: <https://drive.google.com/open?id=0B2tOej64N8xJN2VoNDhYUkt6cUk&authuser=0>
- g. **Prueba de escritorio**
  - La **prueba de escritorio** es una herramienta útil para entender que hace un determinado algoritmo y verificar si el algoritmo cumple con la especificación para lo cual fue pensado.
  - La **prueba de escritorio** se puede considerar una ejecución a mano del algoritmo, por lo tanto debe llevar el registro de los valores que va tomando cada una de las variables involucradas en el mismo.
  - La **prueba de escritorio** se la puede realizar por medio de una matriz cuyas columnas representan a todas las variables involucradas en el proceso. Luego se sigue el flujo del programa.

### Consejos para la prueba de escritorio:

- Incluir todas las variables en la matriz de prueba.
- Seguir el flujo del programa estrictamente.
- Escoger diferentes datos de entrada para validar todas las opciones del algoritmo.
- Es una buena práctica escribir a un costado el símbolo de impresión y mostrar lo que la pantalla va presentando.

**Ejemplo:** prueba de escritorio de algoritmo para el cálculo de la pendiente de una recta.

```

1 x1 = int(input("Ingrese un valor:"))
2 y1 = int(input("Ingrese un valor:"))
3 x2 = int(input("Ingrese un valor:"))
4 y2 = int(input("Ingrese un valor:"))
5
6 r1 = y2-y1
7 r2 = x2-x1
8 m=r1/r2
→ 9 print(m)

```

[Edit code](#)

Program terminated

→ line that has just executed  
 → next line to execute

Program output:

```

Ingrese un valor:4
Ingrese un valor:5
Ingrese un valor:3
Ingrese un valor:1
4.0

```

Frames

Global frame	
x1	4
y1	5
x2	3
y2	1
r1	-4
r2	-1
m	4.0

Figura A1.17: Ejemplo de prueba de escritorio.  
Fuente: (Maldonado, 2013)

## ACTIVIDADES

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	La prueba de escritorio es una forma de representación de un algoritmo.	Verdadero o Falso		Falso
2	La prueba de escritorio es una herramienta que permite conocer el comportamiento de un algoritmo y ver si cumple con los requerimientos deseados.	Verdadero o Falso		Verdadero

Tabla A1.12: Actividades de la lección 7 del módulo 2.  
Fuente: Elaboración propia.

## MATERIAL ADICIONAL

- **Video:** Prueba de Escritorio. Disponible en: <http://youtu.be/oAcu99GERU8>

### Unidad 3: Condicionales y Estructuras de Control

#### a. Secuencia

La estructura de control más simple está representada por una sucesión de operaciones (por ejemplo asignaciones, donde asignar significa almacenar valores constantes o variables), en la que el orden de ejecución coincide con el orden físico de aparición de las instrucciones.



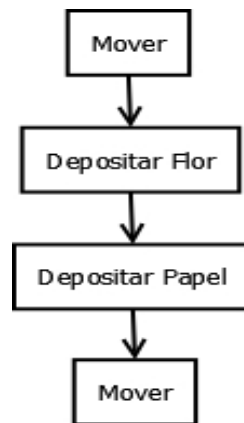


Figura A1.18: Secuencia de instrucciones.  
Fuente: (De Giusti, 2001)

Para la entrada de datos de un programa por teclado se utiliza la función **input**, esta función permite que un programa almacene en una variable lo que escribas en el teclado. Al llegar a la función el programa se detiene esperando que escribas algo y pulses la tecla **Enter**.

#### Ejemplo:

- `print ("Cual es tu nombre")`
- `nombre = input()`
- `print ("Es un gusto conocerte," , nombre)`

Por defecto `input` interpreta lo que se introduce como una cadena caracteres o string, para que se interprete como un valor de un tipo de dato diferente a cadena hay que anteponer el nombre del tipo de datos antes del `input`:

```
valor = int(input("Ingrese un valor"))
```

En este caso `valor` será de tipo entero. A continuación se presentan más ejemplos:

- **Ejemplo 1:** realizar un programa que pida al usuario una cantidad de dólares y muestre su equivalente en euros. Considerar el valor de \$1,4 dólares por Euro.

```
cant= int(input("Ingrese una cantidad de dolares:"))
```

```
print (cant, "dolares son", cant/1.4, "euros")
```

- **Ejemplo 2:** realizar un programa que pida al usuario una cantidad de euros y los transforme a su equivalente en dólares. Considerar que la cantidad introducida puede incluir centavos de euro.

```
cant= float(input("Ingrese una cantidad de euros:"))
```

```
print (cant, " euros son", cant*1.4, " dolares ")
```

- **Ejemplo 3:** suponga que hoy es 3 de abril de 2013, realice un programa que pida al usuario su nombre y año de nacimiento para indicar su edad.

```

nombre = input("Cual es su nombre")

a_naci=int(input("Cual es su año de nacimiento"))

print (nombre, " tienes ", 2013 - a_naci, " años")

```

### ACTIVIDADES

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	Una estructura secuencial es en la que el orden de ejecución puede variar en comparación con el orden físico.	Verdadero o Falso		Falso
2	La función <b>input</b> permite almacenar los datos ingresados por el teclado en una variable.	Verdadero o Falso		Verdadero
3	En la sentencia <b>variable=int(input("ingrese un valor"))</b> , ¿qué tipo de dato almacenará variable?	Opción múltiple con una respuesta	a) Cadena de caracteres b) Booleano c) Entero	c

Tabla A1.13: Actividades de la lección 1 del módulo 3.  
Fuente: Elaboración propia.

### MATERIAL ADICIONAL

- **Video:** Estructura Secuencial. Disponible en: [http://youtu.be/z1b6J-7\\_vuw](http://youtu.be/z1b6J-7_vuw)
- **Lectura:**
  - **Lectura 1;** páginas: 2-3; disponible en: <https://drive.google.com/file/d/0B2tOej64N8xJYU1YNDdsN0JWRm8/edit?usp=sharing>

#### b. Decisión

En un algoritmo representativo de un problema real es prácticamente imposible que las instrucciones sean secuenciales puras. Es necesario tomar decisiones en función de los datos del problema.

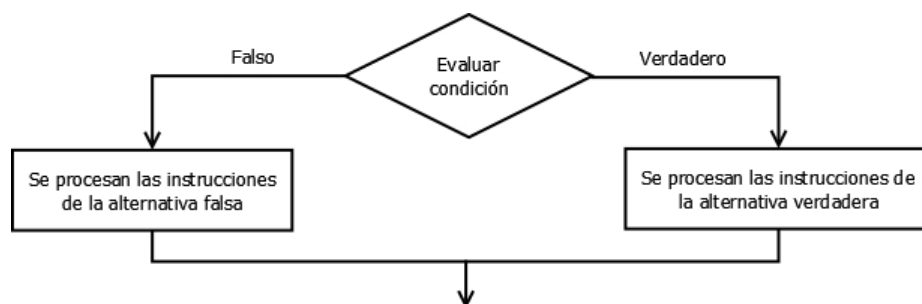


Figura A1.19: Estructura de decisión.  
Fuente: (De Giusti, 2001)

Las sentencias condicionales permiten variar el flujo de ejecución de un programa. El flujo dependerá del valor de una instrucción:

- **If (Simple:** si la expresión es cierta toma un valor específico)
- **If ..else (Doble:** si la expresión es cierta/falsa o toma/no toma un valor específico)

- If...elif...else (**Múltiple**: si la expresión toma un valor u otro, u otro o ninguno de los considerados)

### Sintaxis

**instrucción 1**

**if condición:**

**sentencia**

**instrucción 2**

- Si la condición es cierta, es decir se evalúa a True entonces se ejecuta:

**instrucción 1**

**sentencia**

**instrucción 2**

- Si la condición es falsa se ejecuta:

**instrucción 1**

**instrucción 2**

La evaluación de la condición implica obtener un valor único verdadero o falso como resultado de la evaluación. Por ejemplo:

- **A = 5**

**If (A>10).....**

El resultado de evaluar esta condición es False

- **A = 5**

**If (A<10).....**

El resultado de evaluar esta condición es True

### ¿Qué es una sentencia?

Puede ser una única instrucción o un grupo de instrucciones, pero en este segundo caso éstas deben formar un bloque, es decir, deben estar indexadas respecto a la línea anterior.



Figura A1.20: Instrucciones en bloques.

Fuente: (Maldonado, 2013)

Como se puede observar en la imagen anterior la sentencia estaría formada por la instrucción 1 y la instrucción 2, mientras que la instrucción 3 sería independiente o parte de otra sentencia que ya estaría fuera del bloque if. Ejemplos:



- En este caso se pedirá que se ingrese el número 3 y a través de dos bloques if simples se comprobará si se ingresó el número apropiado:

```
a= int (input("ingrese el numero 3:"))

if a == 3:

    print ("Correcto ingresaste el numero 3")

if a!=3:

    print ("Ingresaste un numero diferente a 3")
```

- En este caso se realizará lo mismo del ejemplo anterior pero esta vez utilizando un **bloque if doble**:

```
a= int(input("ingrese el numero 3:"))

if a == 3:

    print ("Correcto ingresaste el numero 3")

else:

    print ("Ingresaste un numero diferente a 3")
```

También es posible **anidar** varios **bloques if**, permitiéndose ejecutar una y solo una de las sentencias que se encuentren dentro del bloque if en el cuál se cumpla la condición:

```
if condicion1:

    sentencia 1

elif condicion2:

    sentencia2

elif condicion3:

    sentencia 3

else:

    sentencia 4
```

La **condición** se puede crear utilizando los operadores de condición:

operador	comparación
==	es igual que
!=	es distinto de
<	es menor que
<=	es menor o igual que
>	es mayor que
>=	es mayor o igual que

*Figura A1.21: Operadores de comparación.  
Fuente: (De Giusti, 2001)*

A continuación se muestra un ejemplo del uso de **bloques if anidados** y las **condiciones** formadas por diferentes **operadores**:

```

1 x=1
2 y= int(input ("Ingrese un valor"))
3 if (x==y):
4     print ("Son iguales")
5 elif (x>y):
6     print ("X es mayor a Y")
7 elif (x<y):
8     print ("X es menor que Y")

```

Frames

Global frame	
x	1
y	3

Program output:

```

X es menor que Y

```

Figura A1.22: Ejemplo de bloques if anidados y condiciones con operadores.  
Fuente. (Maldonado, 2013)

También se puede hacer uso de los operadores **AND, OR, NOT**:

```

1 x= int(input ("Ingrese un valor PARA X"))
2 y= int(input ("Ingrese un valor PARA Y"))
3 if (x==y) and (x>0):
4     print ("Son iguales y son valores positivos")
5 elif (x>y) and (x>0):
6     print ("X es mayor a Y, X es positivo")
7 elif not(x<y):
8     print ("X es mayor que Y")
9 else:
10    print ("X es menor que Y")

```

Edit code

<< First < Back Program terminated Forward > Last >>

⇒ line that has just executed  
→ next line to execute

Program output:

```

X es menor que Y

```

Figura A1.23: Ejemplo del uso de los operadores AND, OR, NOT.  
Fuente: (Maldonado, 2013)

## Pseudocódigo

- If...Else

```

Si <condicion>
    <sentencia 1>
    <sentencia 2>
    .
    <sentencia n>
Caso Contrario
    <sentencia 1>
    <sentencia 2>
    .
    <sentencia n>
Fin si

```

Figura A1.24: Estructura if-else en pseudocódigo.  
Fuente: (De Giusti, 2001)

- **If...Else Anidados**

```

Si <condicion>
    <sentencia 1>
    <sentencia 2>
    -
    <sentencia n>
Si no entonces Si <condicion>
    <sentencia 1>
    <sentencia 2>
    -
    <sentencia n>

Caso Contrario
    <sentencia 1>
    <sentencia 2>
    -
    <sentencia n>

Fin si
  
```

Figura A1.25: Estructura if-else anidados en pseudocódigo.

Fuente: (De Giusti, 2001)

## ACTIVIDADES

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	Una estructura de decisión permite evaluar cierta condición y decidir cuál es la sentencia indicada a ejecutarse.	Verdadero o Falso		Verdadero
2	Los operadores lógicos <b>and</b> , <b>or</b> y <b>not</b> no pueden ser empleados en la condición.	Verdadero o Falso		Falso
3	¿Cuáles de estos son estructuras de decisión?	Opción múltiple con una respuesta	a) IF... Simple b) IF...ELSE... Doble c) IF...ELIF...ELSE... Múltiple d) Todos los anteriores e) Ninguno de los anteriores	d

Tabla A1.14: Actividades de la lección 2 del módulo 3.

Fuente: Elaboración propia

## MATERIAL ADICIONAL

- **Video:** Tutorial Python ESTRUCTURAS CONDICIONALES. Disponible en: <http://youtu.be/fJLQYoQFGKo>
- **Lectura:**
  - **Lectura 1;** páginas: 6-8; disponible en: <https://drive.google.com/file/d/0B2tOej64N8xJYU1YNDdsN0JWRm8/edit?usp=sharing>
  - **Estándar Pseudocódigo;** página: 2; disponible en: <https://drive.google.com/file/d/0B2tOej64N8xJdDICT2ZOMIN1SGM/edit?usp=sharing>

### c. Repetición

Una extensión natural de la secuencia consiste en repetir  $n$  veces un bloque de acciones.

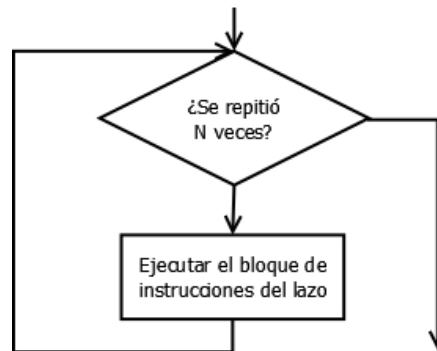


Figura A1.26: Estructura de repetición.

Fuente: (De Giusti, 2001)

Por ejemplo escribamos el algoritmo que me permite mostrar por pantalla la tabla de multiplicar del número 5, utilizando los 6 primeros números enteros.

#### INICIO

```

a = 5

r1 = a * 1
r2 = a * 2
r3 = a * 3
r4 = a * 4
r5 = a * 5
r6 = a * 6

mostrar r1,r2,r3,r4,r5

```

#### FIN

Sin embargo este algoritmo podría volver a escribirse utilizando la estructura de repetición. Para eso observemos lo siguiente:

#### INICIO

a = 5	→ El valor de a contiene el número 5
r1 = a * 1	Los valores por los que se multiplica la variable «a» van en incremento desde 1 hasta 6
r2 = a * 2	
r3 = a * 3	
r4 = a * 4	
r5 = a * 5	
r6 = a * 6	→ Utilizo 6 variables para almacenar el resultado de multiplicar la variable de a * n
mostrar r1,r2,r3,r4,r5	

#### FIN

Figura A1.27: Ejemplo de algoritmo para tabla de multiplicar del 5.

Fuente: (Maldonado, 2013)



Sin embargo este algoritmo podría volver a escribirse utilizando la estructura de repetición. Para eso observemos lo siguiente:

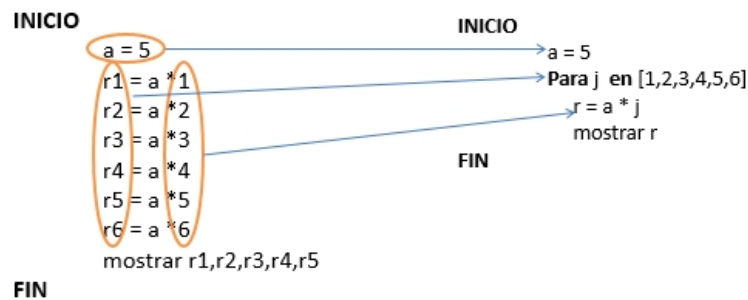


Figura A1.28: Ejemplo de algoritmo utilizando estructuras de repetición.  
Fuente: (Maldonado, 2013)

### Sentencia FOR

Es una repetición condicional que se usa principalmente como repetición fija.

- **Sintaxis:**

**for** variable in serie\_de\_valores:

    instruccion1

**for** variable in serie\_de\_valores:

    instruccion1

    instruccion2

...

- A continuación se muestra la ejecución del algoritmo de la tabla de multiplicar:

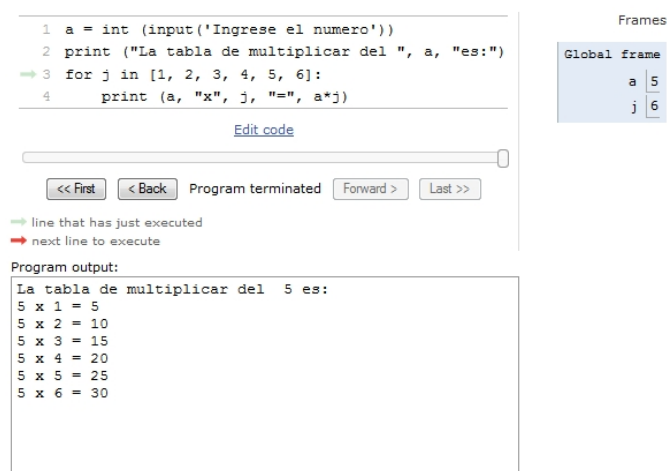


Figura A1.29: Ejecución de algoritmo.  
Fuente: (Maldonado, 2013)

- **Ejemplo 1:**

**for** i in [1,2,3]:

**print** (i)

- **Ejemplo 2:**

```
for nombre in ['Pedro', 'Ana', 'Juan']:
    print("Hola" , nombre)
```

### Función range

La función predefinida **range** permite especificar un rango de valores de manera simple. El uso más simple de la función es con un parámetro: valor final de la serie.

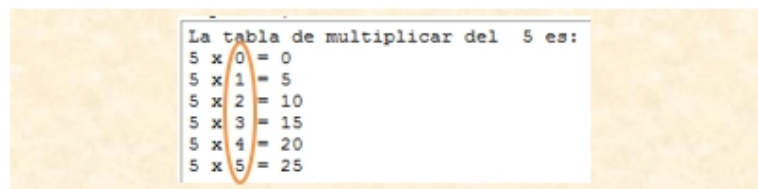
#### Sintaxis:

- **range (valor final)**

El uso anterior equivale a una serie de valores que se inicia en 0 y termina en valor\_final-1, con incrementos de uno en uno.

- **Ejemplo:** realizar un programa que muestre la tabla de multiplicar de un número pedido al usuario.

```
1 a = int (input('Ingrese el numero'))
2 print ("La tabla de multiplicar del ", a, "es:")
→ 3 for j in range (6):
4     print (a, "x", j, "=", a*j)
```



```
La tabla de multiplicar del 5 es:
5 x 0 = 0
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
```

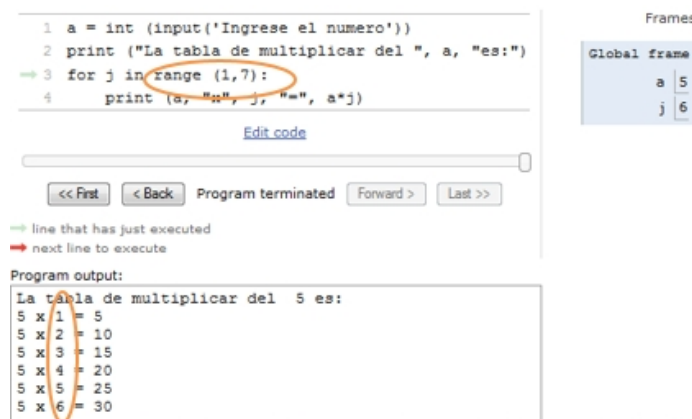
Figura A1.30: Uso de función range.

Fuente: (Maldonado, 2013)

- **range (valor\_inicial, valor\_final)**

El uso anterior equivale a una serie de valores que se inicia en valor\_inicial y termina en valor\_final-1, con incrementos de uno en uno.

- **Ejemplo:** realizar un programa que muestre la tabla de multiplicar de un número pedido al usuario.



```
1 a = int (input('Ingrese el numero'))
2 print ("La tabla de multiplicar del ", a, "es:")
→ 3 for j in range (1,7):
4     print (a, "x", j, "=", a*j)
```

Program output:

```
La tabla de multiplicar del 5 es:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
```

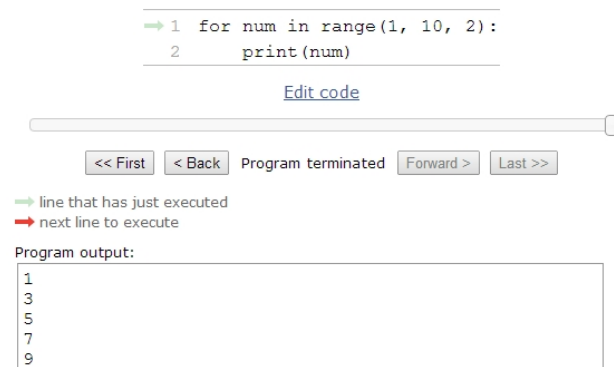
Figura A1.31: Uso de función range con valor inicial y final.

Fuente: (Maldonado, 2013)

- **range (valor\_inicial, valor\_final, incremento/decremento)**

El uso anterior equivale a una serie de valores que se inicia en valor\_inicial y termina en valor\_final-1, con incrementos de «incremento (puede ser: 1, 2, 3, 10, o cualquier entero positivo)» / «decremento (puede ser: -1, -4, -7, o cualquier entero negativo)»

**Ejemplo con incremento:** mostrar números del 1 al 9 pero yendo de dos en dos.



```

1 for num in range(1, 10, 2):
2     print(num)

```

Program output:

```

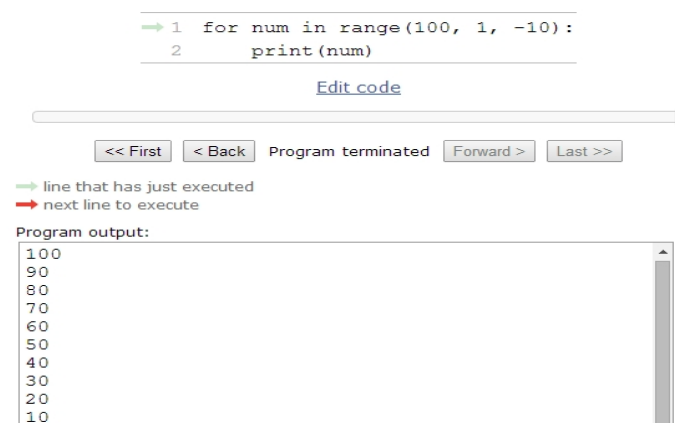
1
3
5
7
9

```

Figura A1.32: Uso de función range con incremento.

Fuente: (Maldonado, 2013)

- **Ejemplo con decremento:** mostrar la serie descendente de números del 100 al 1 pero de 10 en 10.



```

1 for num in range(100, 1, -10):
2     print(num)

```

Program output:

```

100
90
80
70
60
50
40
30
20
10

```

Figura A1.33: Uso de función range con decremento.

Fuente: (Maldonado, 2013)

## Pseudocódigo

- **Estructura FOR con una lista**

```

Para <elemento> en <lista>
    <sentencia 1>
    <sentencia 2>
    .
    .
    <sentencia n>
Fin para

```

Figura A1.34: Pseudocódigo estructura FOR.

Fuente: (De Giusti, 2001)

- **Estructura FOR con la función RANGE**

```

Para <inicialización>,<condición>,<incremento>
    <sentencia 1>
    <sentencia 2>
    .
    .
    <sentencia n>
Fin para
  
```

Figura A1.35: Pseudocódigo estructura FOR con función RANGE.  
Fuente: (De Giusti, 2001)

## ACTIVIDADES

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	Una estructura de repetición permite ejecutar un bloque de instrucciones por n veces.	Verdadero o Falso		Verdadero
2	La función <b>range</b> permite establecer valores de inicio y fin así como el incremento de una estructura <b>for</b> .	Verdadero o Falso		Verdadero
3	¿Cuáles serán los valores de num en la sentencia <b>for num in range (5)</b> ?	Opción múltiple con una respuesta	a) 1,2,3,4 b) 0,1,2,3,4,5 c) 0,1,2,3,4 d) 1,2,3,4,5 e) Ninguna de las anteriores	c
4	¿Cuáles serán los valores de num en la función <b>for num in range (0,8,2)</b> ?	Opción múltiple con una respuesta	a) 1,2,3,4,5,6,7,8 b) 2,4,6,8 c) 0,2,4,6,8 d) 1,3,5,7 e) Ninguna de las anteriores	e

Tabla A1.15: Actividades de la lección 3 del módulo 3.  
Fuente: Elaboración propia

## MATERIAL ADICIONAL

- **Video:** Bucle Para. Disponible en: <http://youtu.be/wyaj6IM89bk>
- **Lectura:**
  - **Lectura 1;** páginas: 3-6; disponible en: <https://drive.google.com/file/d/0B2tOej64N8xJYU1YNDdsN0JWRm8/edit?usp=sharing>
  - **Estándar Pseudocódigo;** páginas: 3-4; disponible en: <https://drive.google.com/file/d/0B2tOej64N8xJdDICT2ZOMIN1SGM/edit?usp=sharing>

#### d. Selección

La estructura de selección se utiliza en el caso de que las alternativas en una estructura básica de decisión sean más de dos.

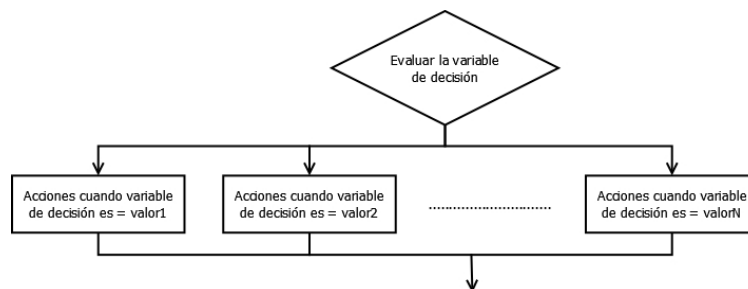


Figura A1.36: Estructura de Selección.  
Fuente: (De Giusti, 2001)

#### Sintaxis

La sintaxis común en los lenguajes de programación

**Select Case** (variable\_decisión)

**Case** (posibilidad1)

sentencia;

**Case** (posibilidad2)

sentencia;

.

.

**Case** (posibilidadN)

sentencia;

**Else**

sentencia;

**End Select**

Sin embargo Python no implementa la estructura Select Case que en otros lenguajes de programación puede ser encontrada como la estructura Switch.

#### ACTIVIDADES

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	La estructura de selección no sirve para evaluar una variable entre varias alternativas.	Verdadero o Falso		Falso
2	El lenguaje de programación Python implementa la estructura de selección.	Verdadero o Falso		Falso

Tabla A1.16: Actividades de la lección 4 del módulo 3.  
Fuente: Elaboración propia.

## MATERIAL ADICIONAL

- **Video:** Lógica de programación - Selección múltiple. Disponible en: <http://youtu.be/3w3Ue1LkPNI>
- **Lectura:**
  - **Lectura 1;** páginas: 8-10; disponible en: <https://drive.google.com/file/d/0B2tOej64N8xJYU1YNDdsN0JWRm8/edit?usp=sharing>

### e. Iteración

Puede ocurrir que se desee ejecutar un bloque de instrucciones desconociendo el número exacto de veces que se ejecutan. Para estos casos en la mayoría de los lenguajes de programación estructurada existen **estructuras de control iterativas condicionales**. Como su nombre lo indica, las acciones se ejecutan dependiendo de la evaluación de la condición definitiva.

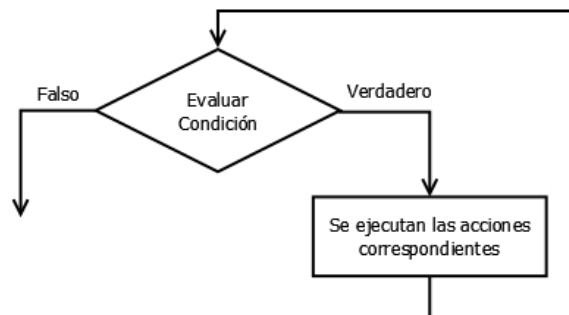


Figura A1.37: Estructura de selección.

Fuente: (De Giusti, 2001)

## Bucles

- Los bucles son estructuras que permiten **ejecutar partes del código de forma repetida** mientras se cumpla una condición.
- Esta **condición** puede ser simple o compuesta de otras condiciones **unidas por operadores lógicos**.
- La instrucción o el bloque de instrucciones que se repiten en el bucle se denominan **cuerpo del bucle**.
- Cada repetición completa de un bucle se denomina **iteración**.

## While

Es una repetición condicional: mientras la condición sea cierta ejecutar el cuerpo del bucle.

- **Sintaxis**

```

while (condición):
    instruccion1
    instruccion2
    instruccion3
    .....
  
```

- **Funcionamiento**

Funcionamiento:

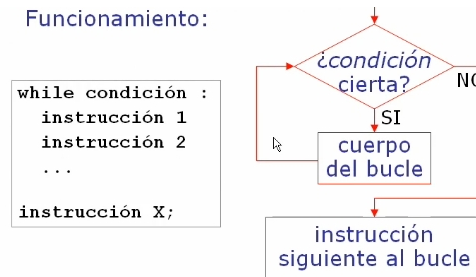


Figura A1.38: Funcionamiento sentencia WHILE.

Fuente: (Maldonado, 2013)

- **Ejemplo 1:** escribir los números del 1 al 10



```

1 contador = 1
2 while (contador <11):
3     print (contador, "-")
4     contador = contador + 1

```

Program output:

```

1 -
2 -
3 -
4 -
5 -
6 -
7 -
8 -
9 -
10 -

```

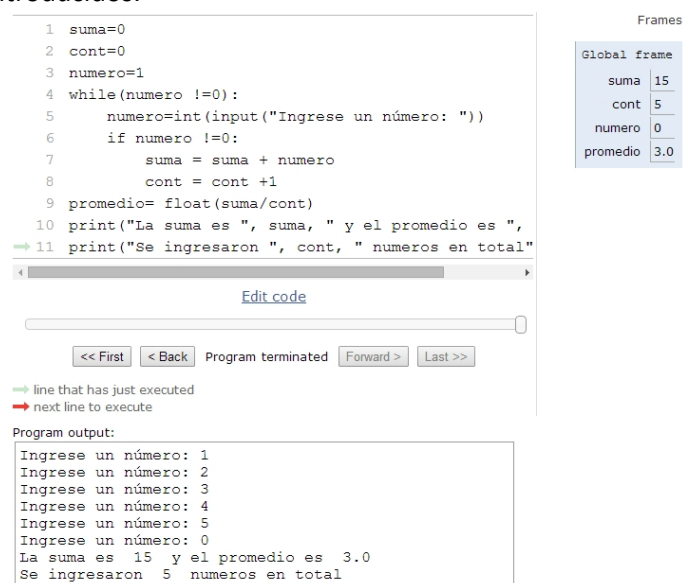
Frames

Global frame	
contador	11

Figura A1.39: Ejemplo 1 de sentencia WHILE.

Fuente: (Maldonado, 2013)

- **Ejemplo 2:** elaborar un programa que vaya pidiendo ingresar números al usuario hasta que se introduzca un 0. Posteriormente debe mostrar la suma y el promedio de todos los números introducidos.



```

1 suma=0
2 cont=0
3 numero=1
4 while(numero !=0):
5     numero=int(input("Ingrese un número: "))
6     if numero !=0:
7         suma = suma + numero
8         cont = cont +1
9     promedio= float(suma/cont)
10    print("La suma es ", suma, " y el promedio es ",
11    print("Se ingresaron ", cont, " numeros en total"

```

Program output:

```

Ingrese un número: 1
Ingrese un número: 2
Ingrese un número: 3
Ingrese un número: 4
Ingrese un número: 5
Ingrese un número: 0
La suma es 15 y el promedio es 3.0
Se ingresaron 5 numeros en total

```

Frames

Global frame	
suma	15
cont	5
numero	0
promedio	3.0

Figura A1.40: Ejemplo 2 de sentencia WHILE.

Fuente: (Maldonado, 2013)





- **Pseudocódigo**

```

Mientras (<condicion>)
    <sentencia 1>
    <sentencia 2>
    .
    <sentencia n>
Fin mientras
  
```

Figura A1.41: Pseudocódigo sentencia WHILE.

Fuente: (De Giusti, 2001)

## ACTIVIDADES

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	Una estructura de iteración puede permitir ejecutar un bloque de instrucciones desconociendo el número exacto de veces que se ejecutarán.	Verdadero o Falso		Verdadero
2	Los bucles son estructuras que permiten ejecutar partes del código de forma repetida mientras se cumpla una condición.	Verdadero o Falso		Verdadero

Tabla A1.17: Actividades de la lección 5 del módulo 3.

Fuente: Elaboración propia.

## MATERIAL ADICIONAL

- **Video:** Tutorial Python 11: Bucles. Disponible en: [http://youtu.be/lyl2ZuOq\\_xQ](http://youtu.be/lyl2ZuOq_xQ)
- **Lectura:**
  - **Lectura 1;** páginas: 11-14; disponible en: <https://drive.google.com/file/d/0B2tOej64N8xJYU1YNDdsN0JWRm8/edit?usp=sharing>

**Estándar Pseudocódigo,** página3. Disponible en:

<https://drive.google.com/file/d/0B2tOej64N8xJdDICT2ZOMIN1SGM/edit?usp=sharing>

## ANEXO 2: BANCO DE PREGUNTAS PARA EVALUACIONES

A continuación detallamos los cuestionarios correspondientes a cada una de las evaluaciones:

### Evaluación Unidad 1

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	La informática es la ciencia que estudia el análisis de problemas utilizando computadoras.	Verdadero o falso		Verdadero
2	Un programa permite resolver un problema mediante instrucciones aleatorias ejecutadas en el ordenador.	Verdadero o falso		Falso
3	Un algoritmo es:	Opción múltiple con una respuesta	1. Secuencia de pasos e instrucciones que producen un resultado deseado y es finito 2. Secuencia de pasos e instrucciones con los cuales no siempre se alcanza el resultado deseado. 3. Secuencia de pasos e instrucciones infinitas. 4. Secuencia de pasos e instrucciones que no siempre alcanzan el resultado deseado	1
4	Las etapas en la resolución de un problema son:	Opción múltiple con una respuesta.	1. Análisis, Especificación, Escritura, Verificación 2. Análisis, Diseño, Especificación, Escritura 3. Análisis, Implementación, Escritura, Verificación 4. Análisis, Diseño, Especificación, Escritura, Verificación	5
5	La descomposición en módulos no es una herramienta útil para resolver un problema.	Verdadero o falso		Falso
6	Un programa es un conjunto de instrucciones. Cuando se escribe por ejemplo las directrices para llegar a la casa de un	Verdadero o falso		Verdadero



	amigo, se está escribiendo un programa.			
7	Una especificación es el proceso de analizar los problemas del mundo real y determinar de forma clara y concreta un objetivo.	Verdadero o falso		Verdadero
8	Python:	Opción múltiple con una respuesta	1. Es un lenguaje de programación interpretado, con tipado dinámico sólo para plataformas Windows. 2. Es un lenguaje de programación con tipado estático multiplataforma. 3. Es un lenguaje de programación interpretado, con tipado dinámico y multiplataforma.	3
9	Especificar un problema significa establecer en forma unívoca su contexto, precondiciones y el resultado esperado.	Verdadero o falso		Verdadero
10	A partir de un problema se debe realizar un proceso de abstracción.	Verdadero o falso		Verdadero

Tabla A2.1: Evaluación del módulo 1.

Fuente: Elaboración propia

## Evaluación Unidad 2

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	Una de las características de los algoritmos es que:	Opción múltiple con una respuesta	1. Tiene inicio y fin 2. Es infinito 3. Es complejo 4. No tiene solución	1
2	Una variable es una estructura que puede ir cambiando su valor a lo largo de la ejecución del programa.	Verdadero o Falso		Verdadero
3	La sentencia que permite la salida de datos por pantalla es:	Completar		print
4	Los tipos de datos se caracterizan por:	Opción múltiple con una respuesta	1. Un rango de valores posibles 2. Puede tomar cualquier valor.	1

			3. Su representación es externa.	
5	La variable de tipo Booleano puede tomar valores de:	Opción múltiple con una respuesta	1. 5,46 2. Hola 3. True o False 4. 2	3
6	La variable de tipo entero se divide en:	Opción múltiple con una respuesta	1. Carácter y Entero 2. Real y carácter 3. Booleano y Real 4. Entero y Real	4
7	Los diagramas de flujo están conformados por figuras conectadas por flechas.	Verdadero o Falso		Verdadero
8	¿Cuáles de estas palabras pertenecen al pseudocódigo?	Opción múltiple con una respuesta	1. INICIO, FIN, capturar, multiplicar 2. INICIO, FIN, leer, escribir 3. EMPEZAR, TERMINAR, capturar, borrar 4. EMPEZAR, TERMINAR, leer, escribir 5. Ninguna de las anteriores	2
9	Los comentarios en un algoritmo son un agregado al programa.	Verdadero o Falso		Falso
10	La prueba de escritorio es una forma de representación de un algoritmo.	Verdadero o Falso		Falso

Tabla A2.2: Evaluación del módulo 2.

Fuente: Elaboración propia

### Evaluación Unidad 3

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	Una estructura secuencial es en la que el orden de ejecución es el mismo en comparación con el orden físico.	Verdadero o Falso		Verdadero
2	En la sentencia <b>variable=int(input("ingrese un valor"))</b> , ¿qué tipo de dato almacenará variable?	Opción múltiple con una respuesta	1. Real 2. Cadena de caracteres 3. Booleano 4. Ninguna de las anteriores	4
3	Los operadores lógicos <b>and</b> , <b>or</b> y <b>not</b> pueden ser empleados para crear una condición.	Verdadero o Falso		Verdadero

4	¿Cuál de estas es una estructura de decisión?	Opción múltiple con una respuesta	1. FOR... 2. IF...ELSE... 3. WHILE... 4. Ninguna de las anteriores	2
5	Una estructura de repetición no permite ejecutar un bloque de instrucciones por n veces.	Verdadero o Falso		Falso
6	La función range permite establecer valores de inicio y fin así como el incremento de una estructura for.	Verdadero o Falso		Verdadero
7	¿Cuáles serán los valores de <b>num</b> en la función <b>for num in range (3,8)</b> ?	Opción múltiple con una respuesta	1. 3,4,5,6,7,8 2. 4,5,6,7 3. 1,2,3,4,5,6,7 4. 3,4,5,6,7 5. Ninguna de las anteriores	4
8	El lenguaje de programación Python no implementa las estructuras de selección.	Verdadero o Falso		Verdadero
9	Los bucles son estructuras que permiten ejecutar partes del código de forma repetida mientras se cumpla una condición.	Verdadero o Falso		Verdadero
10	While es una estructura de selección.	Verdadero o Falso		Falso

Tabla A2.3: Evaluación del módulo 3.

Fuente: Elaboración propia



## Evaluación Final

No.	PREGUNTA	TIPO	OPCIONES	RESPUESTA
1	Un programa es correcto cuando cumple con su especificación, esto significa:	Opción múltiple con una respuesta	1. Cumple con los requerimientos propuestos 2. Está bien documentado. 3. La solución es única 4. Tiene muchas soluciones	1
2	Un lenguaje de programación es muy similar al lenguaje humano.	Verdadero o falso		Falso
3	Los datos de salida son mostrados por la pantalla.	Verdadero o falso		Verdadero
4	A partir del modelo se debe elaborar el análisis de la solución como sistema	Verdadero o falso		Verdadero
5	Una de las ventajas de usar Python:	Opción múltiple con una respuesta	1. Es inexpresivo 2. Es ilegible 3. Tiene entorno estático 4. Es pagado 5. Es gratuito	5
6	Llegar a un modelo significa que es computable	Verdadero o falso		Falso
7	Una variable es una estructura que nunca cambia su valor a lo largo de la ejecución del programa.	Verdadero o falso		Falso
8	¿Cuál de estos es un nombre de variables correcto?	Opción múltiple con una respuesta.	1. bandera1 2. num2 3. numero 4. Todas son correctas 5. Ninguna es correcta	4
9	Los tipos de datos primitivos se dividen en:	Opción múltiple con una respuesta	1. Reales, simples, caracteres 2. Caracteres, numéricos, lógicos 3. Simples, primitivos 4. Lógicos, booleanos	2
10	Escoja un ejemplo de un tipo de dato real:	Opción múltiple con una respuesta.	1. '5.3' 2. 88 3. 7,89	3
11	Los diagramas de flujo son una descripción grafica de un algoritmo para la resolución de problemas.	Verdadero o Falso		Verdadero



12	Un programa bien documentado será más fácil de leer y mantener. Para ello la documentación es fundamental.	Verdadero o Falso		Verdadero
13	Los comentarios no son un agregado al programa, sino una parte importante del mismo.	Verdadero o Falso		Verdadero
14	La prueba de escritorio es una herramienta que permite conocer el comportamiento de un lenguaje de programación.	Verdadero o Falso		Falso
15	Una estructura secuencial es en la que el orden de ejecución puede variar en comparación con el orden físico de las sentencias.	Verdadero o Falso		Falso
16	Los datos de entrada pueden ser capturados por una variable.	Verdadero o Falso		Verdadero
17	Una estructura de decisión permite evaluar cierta condición y decidir cuál es la sentencia indicada a ejecutarse.	Verdadero o Falso		Verdadero
18	La función <b>range</b> permite establecer valores de inicio y fin así como el incremento de una estructura <b>for</b> .	Verdadero o Falso		Verdadero
19	¿Cuáles serán los valores de num en la función <b>for</b> num <b>in range</b> (5, 25, 5)?	Opción múltiple con una respuesta.	1. 1,2,3,4,5 2. 5,10,15,20,25 3. 0,5,10,15,20,25 4. 0,5,10,15,20 5. Ninguna de las anteriores	5
20	While no es una estructura iterativa.	Verdadero o Falso		Falso

Tabla A2.4: Evaluación final

Fuente: Elaboración propia





## ANEXO 3: Herramientas para la implementación de un data warehouse

### Procesos ETL

Los procesos de extracción, transformación y carga (extract, transform, load. ETL) son un proceso que organiza el flujo de los datos entre diferentes sistemas en una organización y aporta métodos y herramientas necesarias para mover datos desde múltiples fuentes a un DW, reformatearlos, limpiarlos y cargarlos en otra base de datos, data mart o almacén de datos.

La idea es que una aplicación ETL lea los datos primarios de alguna fuente de información, realice una transformación, validación, el proceso cualitativo, filtración y al final escriba datos en un DW para que de esta manera los datos queden disponibles para ser analizados por los usuarios.

Un proceso ETL consta de los siguientes pasos:

- **Extraer:** Esta es la primera parte del proceso ETL y consiste en extraerlos datos desde los sistemas de origen. La mayoría de los proyectos de almacenamiento de datos fusionan datos provenientes de diferentes sistemas de origen. La extracción convierte los datos a un formato preparado para iniciar el proceso de transformación.
- **Transformar:** La fase de transformación aplica una serie de reglas de negocio o funciones sobre los datos extraídos para convertirlo en datos que serán cargados. Algunas fuentes de datos requerirán alguna pequeña manipulación de los datos, no obstante en otros casos pueden ser necesarias aplicar algunas de las siguientes transformaciones:
  - Seleccionar sólo ciertas columnas para su carga.
  - Traducir códigos
  - Codificar valores libres
  - Obtener nuevos valores calculados.
  - Unir datos de múltiples fuentes.
  - Calcular totales de múltiples filas de datos.
  - Generación de campos claves en el destino.
  - Transponer columnas
  - Dividir una columna en varias.
- **Cargar:** La fase de carga es el momento en el cual los datos de la fase anterior son cargados en el sistema de destino. Dependiendo de los requerimientos de la organización, este proceso puede abarcar una amplia variedad de acciones diferentes. En algunas bases de datos se sobrescribe la información antigua con nuevos datos. Los data warehouse mantienen un historial de los registros de manera que se pueda hacer una auditoría de los mismos y disponer de un rastro de toda la historia de un valor a lo largo del tiempo. Existen dos formas básicas de desarrollar el proceso de carga:
  - Acumulación simple: Consiste en realizar un resumen de todas las transacciones comprendidas en el período de tiempo seleccionado y transportar el resultado como una única transacción hacia el data warehouse.
  - Rolling: Se aplica en los casos en que se opta por mantener varios niveles de granularidad. Para ello almacena información resumida a distintos niveles, correspondientes a distintas agrupaciones de la unidad de tiempo o diferentes



niveles jerárquicos en alguna o varias de las dimensiones de la magnitud almacenada. (Ramos, 2011)

## Pentaho Data Integration (Kettle)

Kettle es una herramienta de Pentaho Data Integration que permite implementar los procesos de extracción, transformación y carga de datos (ETL). El uso de kettle permite evitar grandes cargas de trabajo manual frecuentemente difícil de mantener y de desplegar. (Ecured, 2014)

Las características de esta herramienta son:

- Open source y sin costes de licencia.
- Entorno gráfico de desarrollo
- Uso de tecnologías estándar: Java, XML, JavaScript.
- Fácil de instalar y configurar.
- Multiplataforma (Windows, Macintosh, Linux).
- Basado en dos tipos de objetos: Transformaciones (colección de pasos en un proceso ETL) y Jobs (Colección de transformaciones).
- Incluye cuatros componentes fundamentales:
  - SPOON: utilizada para diseñar transformaciones ETL usando el entorno gráfico.
  - PAN: para la ejecución de las transformaciones diseñadas en SPOON.
  - CHEF: Para crear Jobs
  - KITCHEN: Para ejecutar Jobs. (Gravitar, 2014)

## JSON

JSON (JavaScript Object Notation – Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para los desarrolladores, mientras que para las máquinas es simple interpretarlo y generarlo. JSON es un formato de texto ligero que es completamente independiente del lenguaje pero utiliza convenciones que son ampliamente conocidos por programadores de la familia de lenguajes C, Java, JavaScript, Perl, Python, etc., lo que hace que sea un lenguaje ideal para el intercambio de datos.

JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales, virtualmente todos los lenguajes de programación los soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras. (Organización JSON, 2011)

## EXCEL

Excel es un software que permite crear tablas, y calcular y analizar datos. Este tipo de software se denomina software de hoja de cálculo.

Excel forma parte de "Office", un conjunto de productos que combina varios tipos de software para crear documentos, hojas de cálculo, presentaciones y para administrar el correo electrónico.



Excel puede ocuparse también como un software de creación de documentación gracias a la inmensa cantidad de características de diseño de tablas y la opción de limitar el área de impresión de un documento. (Microsoft, 2010)

## **BASES DE DATOS**

Una base de datos es una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular.

Entre las principales características que poseen las bases de datos se pueden mencionar las siguientes:

- Independencia lógica y física de los datos.
- Redundancia mínima
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas y optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.
- Acceso a través de lenguajes de programación estándar. (Pérez Valdés, 2007)

## ANEXO 4: HERRAMIENTAS TECNOLÓGICAS UTILIZADAS PARA EL CASO DE ESTUDIO

### Pentaho BI-Server

La Plataforma Pentaho BI provee el soporte y la infraestructura necesarios para crear soluciones de inteligencia empresarial (BI) a problemas de negocios. El marco proporciona los servicios básicos, incluidos autenticación, registro, auditoría, servicios web y motor de reglas. La plataforma también incluye un motor de solución que integra reportes, análisis, tableros de comandos y componentes de minería de datos. El diseño modular y arquitectura basada en plug-in permite a todos o parte de la plataforma para estar embebida en aplicaciones de terceros por los usuarios finales, así como fabricantes de equipos originales.

La aplicación más conocida de la Plataforma Pentaho BI es la BI Server, que funciona como un sistema basado en administración web de informes, el servidor de integración de aplicaciones y un motor de flujo de trabajo ligero (secuencias de acción.) Está diseñado para integrarse fácilmente en cualquier proceso de negocio. (Summan, 2013)

### Configuración de Pentaho BI-SERVER

Pantalla inicial de Pentaho BI-Seerver

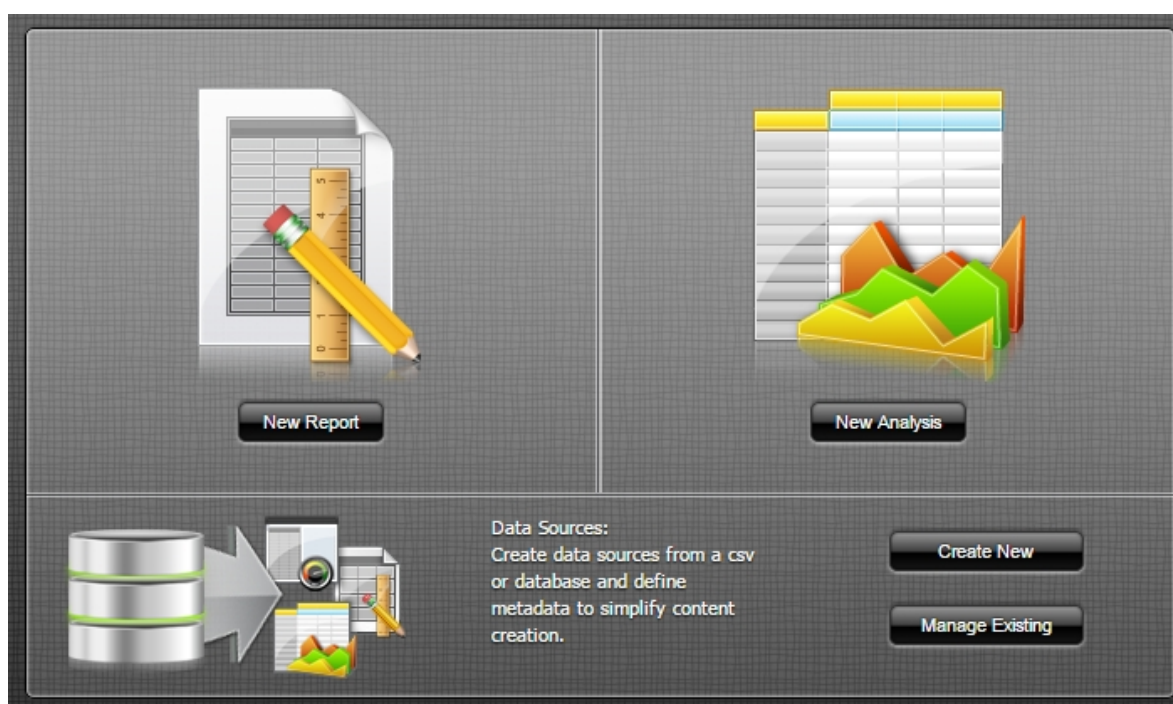


Figura A4.1: Pantalla inicial Pentaho BI-Server.

Fuente: Elaboración propia.

Para poder realizar el análisis se debe crear una fuente de datos, la cual en este caso está ligada a una base de datos. A continuación se muestra la configuración de conexión a la base de datos que contiene el Data Warehouse.

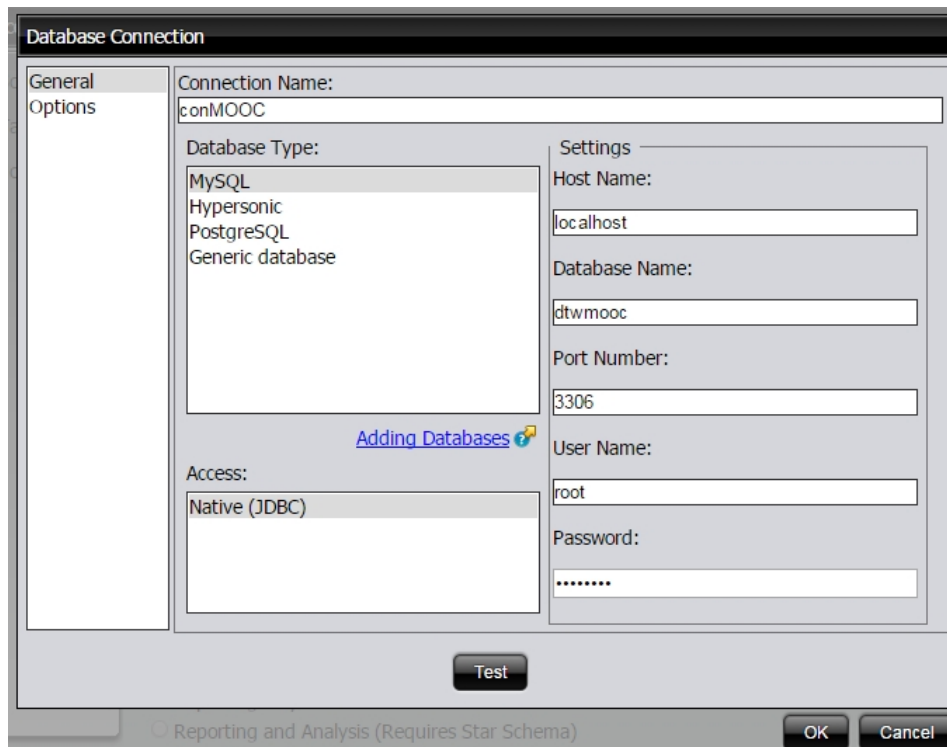


Figura A4.2: Pantalla de configuración de base de datos.

Fuente: Elaboración Propia.

Selección de conexión a base de datos.

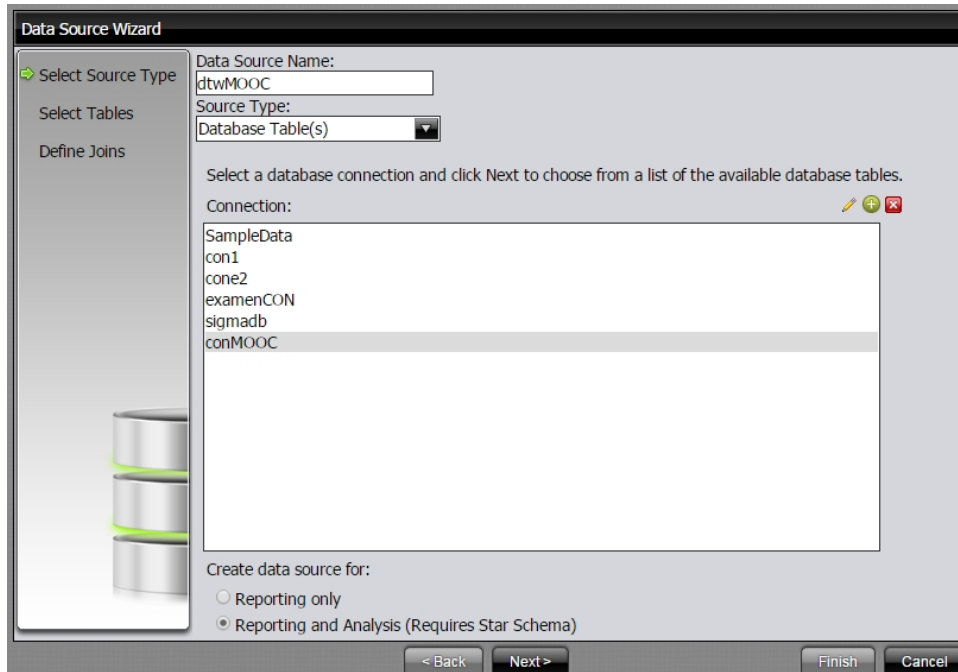


Figura A4.3: Selección de conexión.

Fuente: Elaboración propia.

Selección de tablas que serán parte de las dimensiones del Data Warehouse, y selección de la tabla que se usará como tabla de hechos.

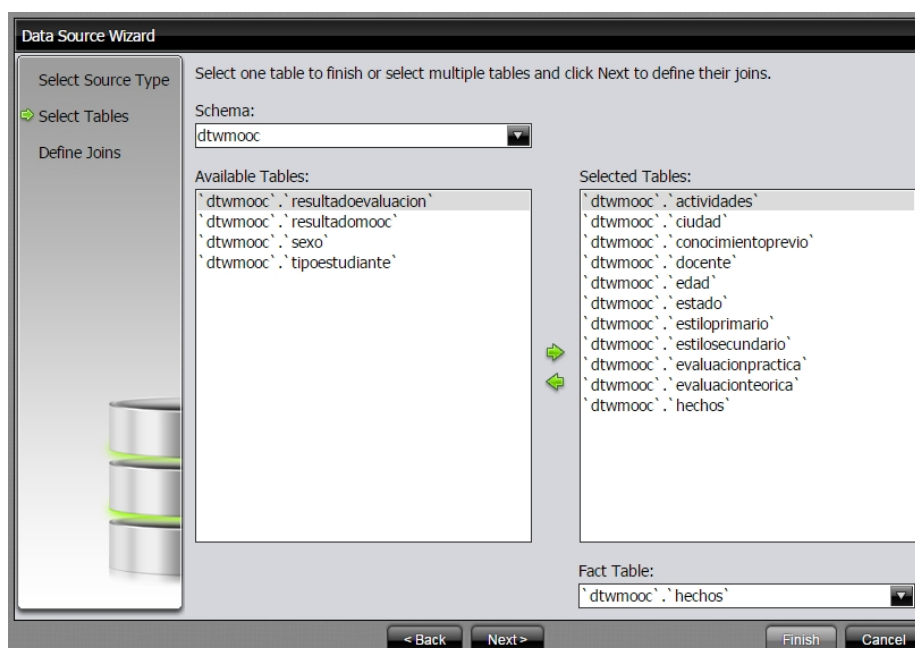


Figura A4.4: Selección de tablas del Data Warehouse

Fuente: Elaboración propia

Selección de tablas y campos para realizar los respectivos joins entre las tablas de dimensiones y la tabla de hechos:

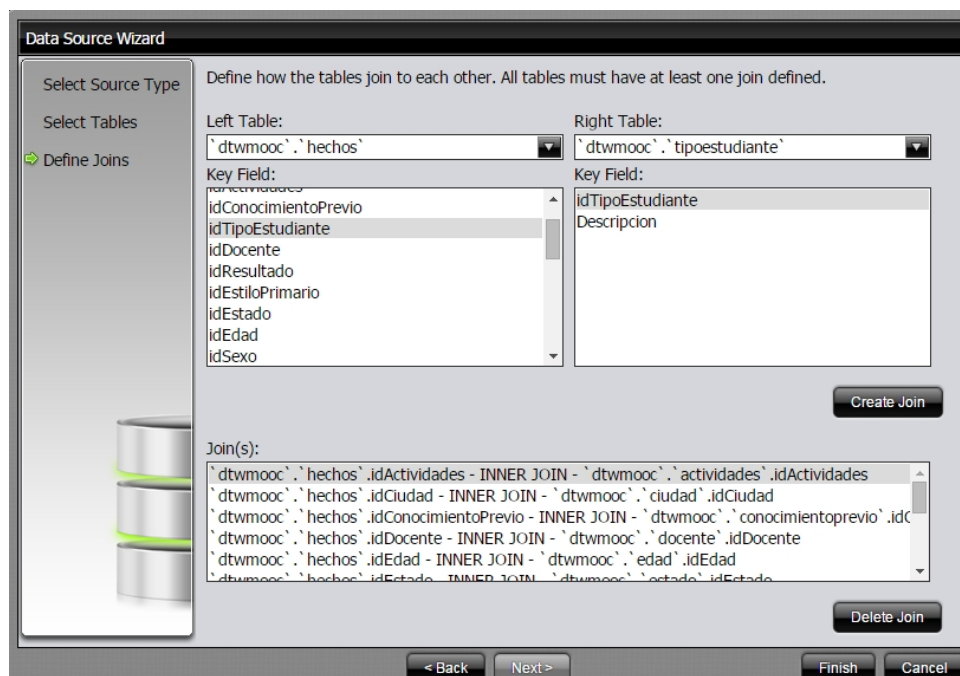


Figura A4. 5: Creación de uniones de tablas.

Fuente: Elaboración propia.

Pentaho BI-Server indica que se crea un cubo multidimensional por defecto, el cual puede ser modificado.

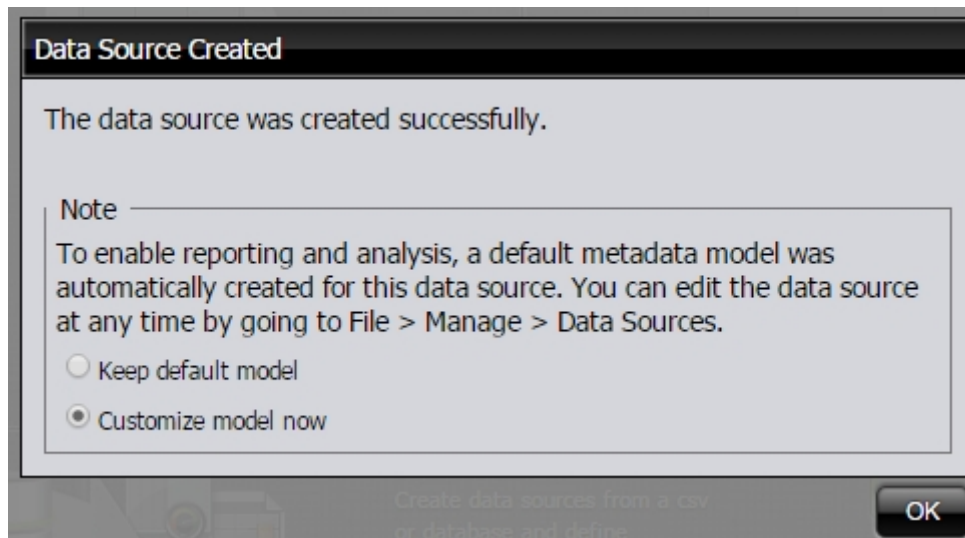


Figura A4.6: Pantalla de Finalización de creación de Data Source.  
Fuente: Elaboración propia.

## Weka

WEKA es una herramienta de aprendizaje automático y Datamining, escrita en lenguaje Java, gratuita y desarrollada en la Universidad de Waikato, Nueva Zelanda (WEKA = Waikato Enviroment for Knowledge Analysis). (Programa de doctorado Tecnologías Industriales, 2013)

El paquete de WEKA contiene una colección de herramientas de visualización y algoritmos para análisis de datos y modelado predictivo, unidos a una interfaz gráfica de usuario para acceder fácilmente a sus funcionalidades.

Las principales características de WEKA son:

- Está disponible libremente bajo la licencia pública general de GNU.
- Es portable porque está completamente implementado en Java y puede correr en la mayoría de plataformas.
- Contiene una extensa colección de técnicas para pre procesamiento de datos y modelado.
- Es fácil de utilizar gracias a su interfaz gráfica de usuario. (Wikipedia, 2014)

La interfaz de usuario principal de WEKA nos ofrece cuatro posibles opciones para acceder a las funcionalidades del programa las mismas que se describen a continuación:

- Simple CLI: es la abreviatura de *Simple Command-Line Interface* (interfaz simple de línea de comandos). Permite acceder a todas las opciones de WEKA desde línea de comandos.
- Explorer: Dispone de varios paneles que dan acceso a los componentes principales del banco de trabajo.
  - El panel "Preprocess" dispone de opciones para importar datos de una base de datos, de un fichero de Excel, CSV, etc., para pre procesar estos datos utilizando algoritmos de filtrado.
  - El panel "Classify" permite al usuario aplicar algoritmos de clasificación estadística y análisis de regresión a los conjuntos de datos resultantes, para





estimar la exactitud del modelo predictivo resultante, y para visualizar predicciones erróneas, curvas ROC, etc.

- El panel “Associate” proporciona acceso a las reglas de asociación aprendidas que intentan identificar todas las interrelaciones importantes entre los atributos.
- El panel “Clúster” da acceso a las técnicas de clustering o agrupamiento de WEKA entre los que podemos mencionar: K-means, CobWeb, DEScan, etc.
- El panel “Selected attributes” proporciona algoritmos para identificar los atributos más predictivos en un conjunto de datos.
- El panel “Visualize” muestra una matriz de puntos dispersos donde cada punto individual puede seleccionarse y agrandarse para ser analizados en detalle usando varios operadores de selección.
- **Experimenter:** Permite la comparación sistemática de una ejecución de los algoritmos predictivos de WEKA sobre una colección de conjuntos de datos.

**Knowledge Flow:** Es una interfaz que en esencia implementa las mismas funciones que Explorer, pero que además ofrece aprendizaje incremental. (Sánchez, 2011)